

2019年11月29日(金) RIST東京事務所



第3回LAMMPS入門講習会

高度情報科学技術研究機構(RIST)

利用支援部 吉澤香奈子

Outline

- MateriApps LIVE!の設定
- LAMMPSについて
 - 力場(ポテンシャル)について
 - 実行時エラー
- LAMMPSのexamplesの実行
 - melt, micelle, colloid を実行する
 - 計算結果をVMDで可視化する
 - micelle と colloid を3次元化する
- サイズを大きくした計算
- restartを行う
- HPCIにプリインストールされたアプリ情報
- 付録

MateriApps LIVE! とは？

- MateriApps LIVE! ウェブサイト
<http://cmisi.github.io/MateriAppsLive/>
- 仮想マシン(VirtualBox)上で直接ブートできるDebian Linux
 - Windows、Macなどで利用可
 - インストール作業なしで物質科学アプリを実行できる
- 最新版: バージョン2.3 (2019年5月14日公開)
本講習会では、最新版を用いる。
- MateriAppsで紹介している公開アプリ・ツールを収録
 - abinit, AkaiKKR, ALPS, CP2K, Feram ,ERmod, DCORE, DSQSS, HΦ, LAMMPS, mVMC, OpenMX, Quantum ESPRESSO, SMASH, xTAPP 等
 - ParaView, Tapioca, VESTA, VMD, XCrysDen...
 - GAMESS, VMDには自動インストーラーを準備
- MateriApps LIVE! サイトからダウンロード可能
 - 2013年7月以来、4500+コピーを配布

必要なファイルのダウンロード

- VirtualBox インストーラ: VirtualBox-*-OSX.dmg, VirtualBox-*-Win.exe
(<https://www.virtualbox.org/wiki/Downloads> からダウンロード可)
- MateriApps LIVE! VirtualBox 仮想ディスクイメージ: MateriAppsLive-*-.ova
(<http://sourceforge.net/projects/materiappslive/files/> からダウンロード可)
- ドキュメント
README.html, README-en.html
[https://github.com/cm*si*/MateriAppsLive/wiki/MateriAppsLive-.ova](https://github.com/cm<i>si</i>/MateriAppsLive/wiki/MateriAppsLive-.ova)

VirtualBox からの起動方法

- ✓ USB メモリのファイルをパソコンに差し込む
- ✓ インストーラをダブルクリックして VirtualBox をインストール
 - Windows版: [VirutalBox-*-Win.exe](#)
 - Mac版: [VirtualBox-*-OSX.dmg](#)
 - 途中の質問には適当に答える
- ✓ MateriApps LIVE! のインポート
 - [MateriAppsLive-2.3-amd64.ova](#) をダブルクリック
 - VirtualBox が起動してインポート画面が開くので、「インポート」ボタンを押す
 - 2~3分かかるが完了するとマネージャーが起動
- ホスト (ホストOS) : もともと動いている OS (Windows、Mac OS X など)のこと
- 仮想マシン (ゲストOS) : VirtualBox の中で動いている OS (= MateriApps LIVE!)

VirtualBox の設定

✓ 設定: 不要なポップアップメッセージを非表示にする

- Windows: USBメモリからコピーした `vbconfig.bat` をダブルクリック

- Mac OS X: `vbconfig.command` をダブルクリック

あるいはターミナルで「`sh vbconfig.command`」を実行

✓ 設定: ホストOSのディスクに仮想マシンからアクセスできるように

1. VirtualBox マネージャー画面で `MateriAppsLive-*` を選択し「設定」

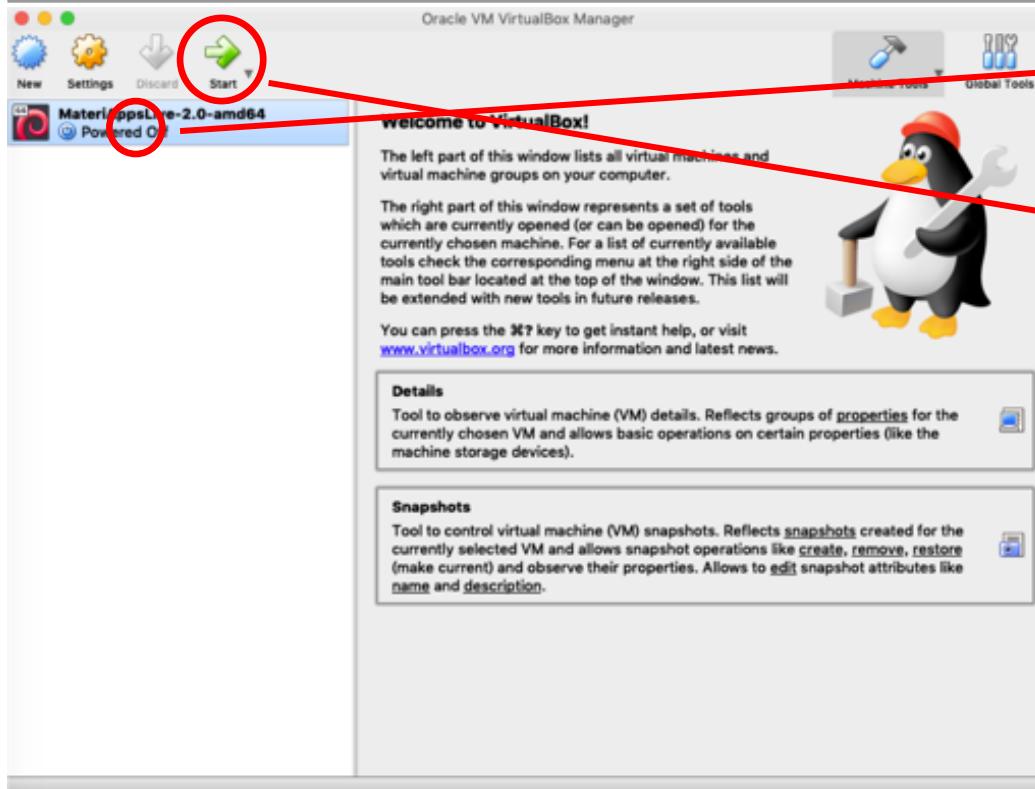
2. 「共有フォルダー」タブを開き、右側の「+」(新規共有フォルダーを追加します)をクリック

3. 「フォルダーのパス」の右側の「v」マークをクリックし、「その他」を選択。さきほどUSBメモリからコピーしたフォルダーを選択する

4. 「自動マウント」をチェックし「OK」⇒さらに「OK」

5. 仮想マシンを起動すると、3で選択したフォルダが、`/media/sf_...` の下に見える

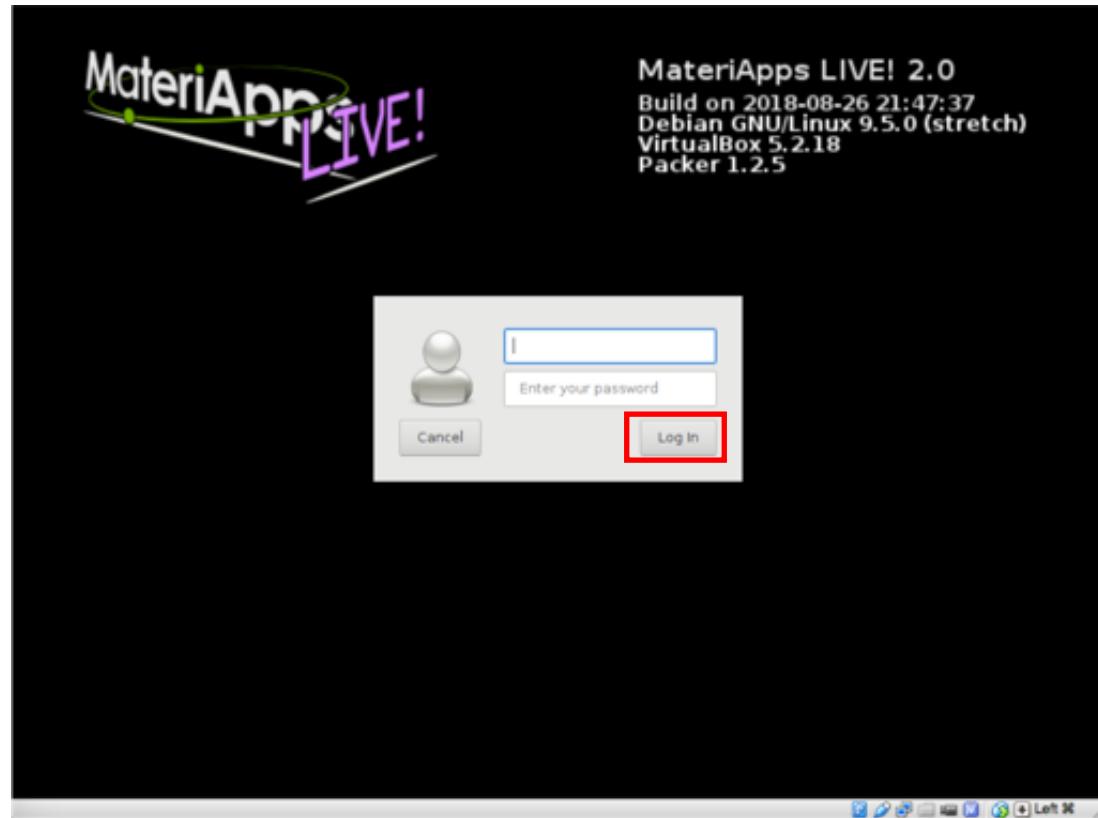
VirtualBoxからの起動



1. 「MateriAppsLive...」を選択
2. 起動ボタンを押す
3. ログイン画面ができるまでそのまま待つ

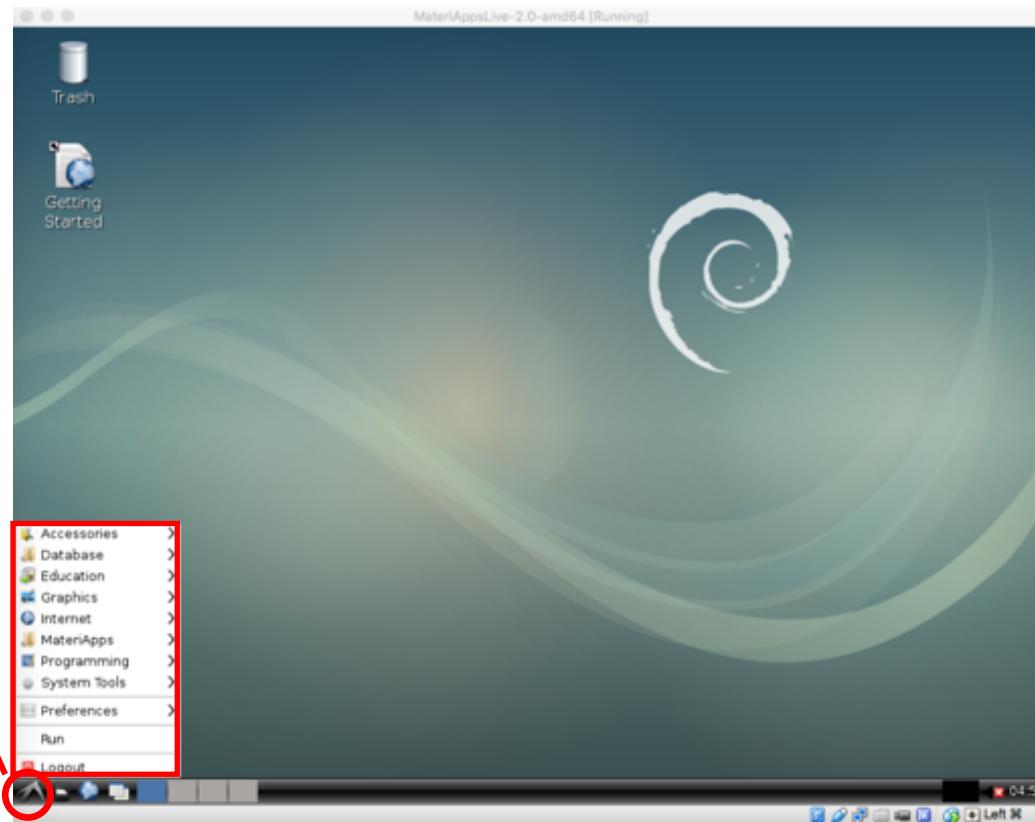
MateriApps LIVE!へのログイン

- しばらくすると下の画面のようなログイン画面が表示される
- 下記の情報を使ってログイン
 - ユーザ名(login): *user*
 - パスワード(password): *live*



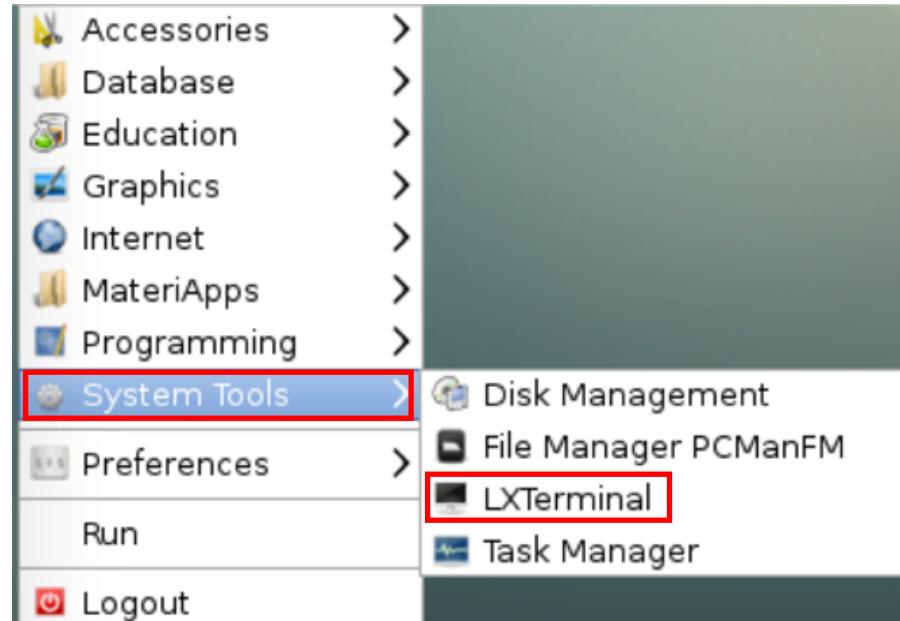
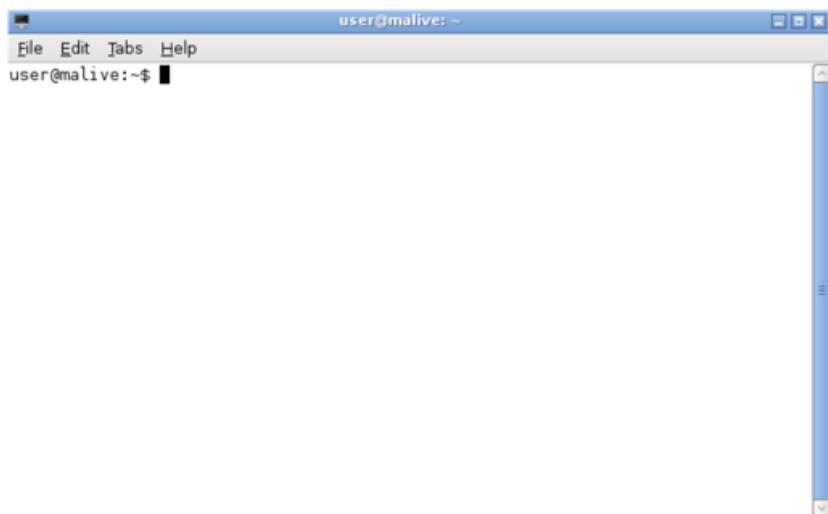
スタートメニュー

- スタートメニュー



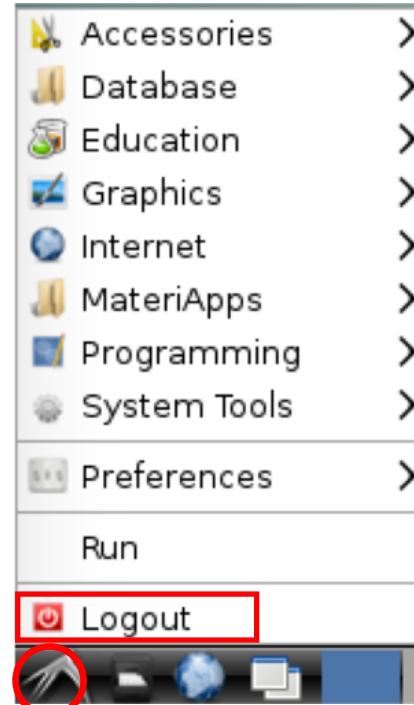
ターミナルの起動

- 「スタートメニュー」
⇒ 「System Tools」
⇒ 「LXTerminal」



MateriApps LIVE! の終了

- ・「スタートメニュー」
⇒ 「Logout」
- ・下の画面の「Shutdown」選択



日本語キーボード、コピー&ペースト

- ・日本語キーボード(「@」が「P」の右にあるタイプ)では、記号が正しく入力できません。その場合、以下の設定を行ってください
 - ・「スタートメニュー」⇒「System Tools」⇒「LXTerminal」
 - ・ターミナル(端末)が立ち上がる所以「`setxkbmap -layout jp`」と入力しリターン
 - ・「@」が正しく入力できることを確認
 - ・(英語配列に戻したいとき: 「`setxkbmap -layout us`」)
- ・ホストOSでPDFファイルからコピーした文字列を、仮想マシンの端末でペーストする方法
 - ・端末上で右クリック ⇒ 「Paste」
 - ・あるいは、「shift」と「control」を同時に押しながら「V」
 - ・文字列のコピーは、右クリック ⇒ 「Copy」あるいは「shift + control + C」

LAMMPSについて

- Large-scale **A**tomic/**M**olecular **M**assively **P**arallel **S**imulator
<http://lammps.sandia.gov/>
- 並列計算機のために設計された分子動力学シミュレータ
 - 物質系を構成する原子や分子の1つ1つに対する運動方程式を数値的に計算している
 - 高並列計算のため2007年にC++でオブジェクト指向で完全に、rewriteされている
オブジェクト指向 → コード拡張・公開 → 多分野・多スケールに渡る計算
 - 現在、粒子系統合シミュレータとして高い支持
- ソースコードダウンロード
<http://lammps.sandia.gov/download.html>
 - 最新のStable version (7 Aug 2019) : **2019年11月15日現在**
ソースファイル名 : **lammps-stable.tar.gz**
 - Stable以外の最新のコードもある
過去のメジャーなコード : <http://lammps.sandia.gov/tars/>
- ドキュメント
<http://lammps.sandia.gov/doc/Manual.html>

何か不明なことがあったらこの
ページを見る

力場(ポテンシャル)について(1)

- LAMMPS Force Fields
http://lammps.sandia.gov/doc/99/force_fields.html
- まずは、パッケージにある examples や potentials を参考にする。
(例) LAMMPSのパッケージ(lammps-stable.tar.gz)の
lammps-11Aug17/potentials
を参考にする。
- 目安としては、計算対象の物性が正しいものを選ぶ。
(例) 常温常圧の鉄は、体心立方格子構造(bcc構造)
lammps-11Aug17/potentials/Fe_mm.eam.fs を用いる

```
# Interatomic potential – Embedded Atom Method
pair style eam/fs
# interaction pairs , filename , Element parameters
pair coeff * * Fe mm.eam. fs Fe
```

入力ファイルの書式例

- 自分でカスタマイズできるが、初心者には難しい。
(→ 専門家に相談)
- 力場(ポテンシャル)にこだわっても、結局は古典計算?
経験的に決める
→ 過去の成功例、論文になっているポテンシャルを用いる。
(非経験的に決めるなら第一原理計算)

物性が正しいように決める

力場(ポテンシャル)について(2)



- ポテンシャルのデータベース
Interatomic Potentials Repository Project
<https://www.ctcms.nist.gov/potentials/>
(例) Al-Ni : <https://www.ctcms.nist.gov/potentials/Al-Ni.html>
Mishin-Ni-Al-2009.eam.alloy を用いる
- 分子モデリングソフトウェアで設定する
下記のURLなどを参考にする
Winmostar: https://winmostar.com/jp/manual_jp.html
Molby: <http://molby.osdn.jp/doc/ja/index.html>

実行時エラー(1)

- 実行時にエラーが出た場合

ホームページ

<https://lammps.sandia.gov/doc/Errors.html>

13.3. Error messages

1-3 bond count is inconsistent

An inconsistency was detected when computing the number of 1-3 neighbors for each atom. This likely means something is wrong with the bond topologies you have defined.

1-4 bond count is inconsistent

An inconsistency was detected when computing the number of 1-4 neighbors for each atom. This likely means something is wrong with the bond topologies you have defined.

Accelerator sharing is not currently supported on system

Multiple MPI processes cannot share the accelerator on your system. For NVIDIA GPUs, see the nvidia-smi command to change this setting.

All angle coeffs are not set

All angle coefficients must be set in the data file or by the angle_coeff command before running a simulation.

...

を参考にする。

実行時エラー(2)

- 経験を積むとエラーの原因がわかりやすくなる。
- (例) colloidのパッケージが入っていない場合
examples の colloid を実行しようとすると、
ERROR: Unknown pair style colloid (../force.cpp:246)
Last command: pair_style colloid 12.5
のエラーが出る。

入力ファイル(in.colloid)において、
“pair_style colloid 12.5” という指定が許されていない

https://lammps.sandia.gov/doc/Errors_messages.html

13.3. Error messages

のページには、

Unknown pair style

The choice of pair style is unknown.

→ make yes-colloid によって、colloidのパッケージを入れれば解決する。
(パッケージのインストールについては、付録を参照)

LAMMPSのexamplesについて

➤ LAMMPS の examples

LAMMPSパッケージ: `lammps-11Aug17/examples`

(MateriApps LIVE!: `/usr/share/doc/lammps-doc/examples`)

➤ examples の解説

<https://lammps.sandia.gov/doc/Examples.html>

• 2次元系(ビジュアル重視)

- `colloid`, `crack`, `flow`, `friction`,
`micelle`, `nemd`, `obstacle`, `shear`

• 3次元系(簡単なもの)

- `melt`, `peptide`

• 力場

- `dreiding`, `kim`, `reax`

• 高速化機能

- `cuda`, `gpu`, `intel`

高分子物理向けのExamples

例	説明
<code>melt</code>	rapid melt of 3d LJ system
<code>micelle</code>	self-assembly of small lipid-like molecules into 2d bilayers
<code>colloid</code>	big colloid particles in a small particle solvent, 2d system
<code>peptide</code>	dynamics of a small solvated peptide chain (5-mer)

高分子物理向けのexamples

- melt: FCC格子状に並んだ LJ 粒子が溶けて拡散
- micelle: 3要素からなるミセル系。会合していく
2次元の計算 (\rightarrow 3次元への拡張を考える)
- colloid: 大きいコロイド粒子と小さい LJ 粒子の系
2次元の計算 (\rightarrow 3次元への拡張を考える)
※ COLLOID package のインストールが必要
(本日の講習では、インストールされているもの
を用いる。)

examples/melt(1)

melt: FCC格子状に並んだ LJ粒子が溶けて拡散

入力ファイル: in.melt

```
# 3d Lennard-Jones melt
```

units lj
atom_style atomic

neigh_modify では、どのくらいの頻度でneighbor listを更新するかの指定。
ここでは、neighbor listの最後の構成から0ステップまで再構成をせず、0ステップから20ステップ後にneighbor listの再構成を行う。

Lennard-Jones (LJ)
古典粒子

lattice fcc 0.8442
region box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass 1 1.0

velocity all create 3.0 87287 速度の設定

pair_style lj/cut 2.5

pair_coeff 1 1 1.0 1.0 2.5

力場パラメータの設定 (ljポテンシャルを指定)

neighbor 0.3 bin
neigh_modify every 20 delay 0 check no

neighbor では、ポテンシャルcutoffよりどれだけ外側までを隣接原子として探すかを指定 (距離内にある粒子で neighbor list を構成する)。ここでは (cutoff + 0.3)Ang の範囲を隣接とし、bin で O(N) の探索手法を用いる

fix

1 all nve

fixで系の制御

NVEアンサンブル

examples/melt(2)

入力ファイル: in.melt (続き)

```
#dump id all atom 50 dump.melt
```

```
#dump 2 all image 25 image.*.jpg type type &
#      axes yes 0.8 0.02 view 60 -30
#dump_modify 2 pad 3
```

は、コメントアウト
dump: 出力形式の指定

```
#dump 3 all movie 25 movie.mpg type type &
#      axes yes 0.8 0.02 view 60 -30
#dump_modify 3 pad 3
```

```
thermo 50
run 250
```

何ステップ毎に温度情報
を出力するかの指定

計算実行(step)数

実行準備

- ホームディレクトリに作業ディレクトリを作る。(例: \$HOME/lammps)
\$ mkdir lammps
- 作成した作業ディレクトリへ移動する。(\$HOME/lammps)
\$ cd lammps
- exmapleファイルのコピー (melt, micelle, colloid)
\$ cp -r /usr/share/lammps/examples/melt ./
\$ cp -r /usr/share/lammps/examples/examples/micelle ./
\$ cp -r /usr/share/lammps/examples/examples/colloid/ ./

meltの実行(1)

- melt へ移動

```
$ cd melt
```

- melt の中のファイルの確認

```
$ ls
```

```
in.melt  log.5Oct16.melt.g++.1  log.5Oct16.melt.g++.4
```

- in.melt の実行

```
$ lammmps < in.melt
```

```
LAMMPS (11 Aug 2017)
```

```
OMP_NUM_THREADS environment is not set. Defaulting to 1 thread.
```

```
(../comm.cpp:90)
```

```
using 1 OpenMP thread(s) per MPI task
```

```
...
```

```
Total # of neighbors = 151513
```

```
Ave neights/atom = 37.8783
```

```
Neighbor list builds = 12
```

```
Dangerous builds not checked
```

```
Total wall time: 0:00:00
```

エラーメッセージなしで
Total wall time が表示されれば
正常終了

- log.lammps というログファイルも出力される。

meltの実行(2)



- MPIプロセス数=4 での実行

```
$ mpirun -n 4 lammps < in.melt  
LAMMPS (11 Aug 2017)
```

...

1 by 2 by 2 MPI processor grid

...

Performance: 90089.282 tau/day, 208.540 timesteps/s

22.1% CPU use with 4 MPI tasks x 1 OpenMP threads

...

マシンに4コア以上ある場合

meltの実行(3)



- スレッド数=4での実行

```
$ export OMP_NUM_THREADS=4
```

```
$ echo $OMP_NUM_THREADS
```

```
4
```

```
$ lammps < in.melt
```

```
LAMMPS (11 Aug 2017)
```

```
using 4 OpenMP thread(s) per MPI task
```

```
...
```

```
Performance: 115561.777 tau/day, 267.504 timesteps/s
```

```
97.6% CPU use with 1 MPI tasks x 4 OpenMP threads
```

```
...
```

マシンに4コア以上ある場合

(補足) OMP_NUM_THREADS指定の削除

```
$ unset OMP_NUM_THREADS
```

```
$ echo $OMP_NUM_THREADS
```

micelle と colloid の実行

- micelle へ移動 → melt と同様の操作
\$ cd \$HOME/lammps/micelle
\$ gzip -d data.micelle.gz
\$ lammps < in.micelle
(注) MateriApps LIVE!では、ファイルが "gz" になっているものがあるので、その時は解凍してから使う。
- colloid へ移動 → melt, micelle と同様の操作
\$ cd \$HOME/lammps/colloid
\$ lammps < in.colloid

VMDで可視化する(1)

- LAMMPSでdumpファイルの出力 → 入力ファイルの編集
- \$HOME/lammpas/melt へ移動
\$ cd \$HOME/lammpas/melt
- in.melt のバックアップを取る
\$ cp in.melt in.melt.org : (例)
- melt の入力ファイルをエディタ(emacs)で編集
\$ emacs in.melt &

VMDで可視化する(2)

- emacs の起動

emacs@malive.local

File Edit Options Buffers Tools Help ?

3d Lennard-Jones melt

```

units      lj
atom_style atomic

lattice    fcc 0.8442
region     box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass       1 1.0

velocity   all create 3.0 87287

pair_style lj/cut 2.5
pair_coeff 1 1 1.0 1.0 2.5

neighbor   0.3 bin

```

in.melt Top L1 (Fundamental)-----

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

[Emacs Tutorial](#) Learn basic keystroke commands
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info
[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
[Copying Conditions](#) Conditions for redistributing and changing Emacs
[Ordering Manuals](#) Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

This is GNU Emacs 23.4.1 (i486-pc-linux-gnu, GTK+ Version 2.24.10)
of 2012-09-10 on murphy, modified by Debian
Copyright (C) 2012 Free Software Foundation, Inc.

Dismiss this startup screen Never show it again.

-U:%%- *GNU Emacs* All L3 (Fundamental)-----

For information about GNU Emacs and the GNU system, type C-h C-a.

①

Welcome to GNU Emacs, one component of the GNU/Linux operating system.

[Emacs Tutorial](#) Learn basic keystroke commands
[Emacs Guided Tour](#) Overview of Emacs features at gnu.org
[View Emacs Manual](#) View the Emacs manual using Info
[Absence of Warranty](#) GNU Emacs comes with ABSOLUTELY NO WARRANTY
[Copying Conditions](#) Conditions for redistributing and changing Emacs
[Ordering Manuals](#) Purchasing printed copies of manuals
To quit a partially entered command, type Control-g.

This is GNU Emacs 23.4.1 (i486-pc-linux-gnu, GTK+ Version 2.24.10)
of 2012-09-10 on murphy, modified by Debian
Copyright (C) 2012 Free Software Foundation, Inc.

Dismiss this startup screen Never show it again.

② ここ(青字)をクリック

① チェックを入れる

②

emacs@malive.local

File Edit Options Buffers Tools Help ?

3d Lennard-Jones melt

```

units      lj
atom_style atomic

lattice    fcc 0.8442
region     box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass       1 1.0

velocity   all create 3.0 87287

pair_style lj/cut 2.5
pair_coeff 1 1 1.0 1.0 2.5

neighbor   0.3 bin
neigh_modify every 20 delay 0 check no

fix        1 all nve

#dump      id all atom 50 dump.melt

#dump      2 all image 25 image.*.jpg type type &
#           axes yes 0.8 0.02 view 60 -30
#           2 pad 3

#dump      3 all movie 25 movie.mpg type type &
#           axes yes 0.8 0.02 view 60 -30
#           3 pad 3

thermo    50
run       250

```

in.melt All L1 (Fundamental)-----

Wrote /home/user/.emacs

③

VMDで可視化する(3)

- melt の入力ファイル(in.melt)の編集

```
# 3d Lennard-Jones melt

units          lj
atom_style    atomic

lattice        fcc 0.8442
region         box block 0 10 0 10 0 10
create_box     1 box
create_atoms   1 box
mass           1 1.0

velocity       all create 3.0 87287

pair_style     lj/cut 2.5
pair_coeff    1 1 1.0 1.0 2.5

neighbor       0.3 bin
neigh_modify   every 20 delay 0 check no

fix            1 all nve

dump          id all atom 50 dump.melt ← #を取る
#dump          2 all image 25 image.*.jpg type type &
#               axes yes 0.8 0.02 view 60 -30
#dump_modify  2 pad 3

#dump          3 all movie 25 movie.mpg type type &
#               axes yes 0.8 0.02 view 60 -30
#dump_modify  3 pad 3

thermo        50
run           250
```

VMDで可視化する(4)

- 保存して、emacs を閉じる

emacs@malive.local

File Edit Options Buffers Tools Help

3d Lennard-Jones melt

units lj
atom_style atomic

lattice fcc 0.8442
region box block 0 10 0 10 0 10
create_box 1 box
create_atoms 1 box
mass 1 1.0

velocity all create 3.0 87287

pair_style lj/cut 2.5
pair_coeff 1 1 1.0 1.0 2.5

neighbor 0.3 bin
neigh_modify every 20 delay 0 check no

fix 1 all nve

#dump id all atom 50 dump.melt

#dump 2 all image 25 image.*.jpg type type &
axes yes 0.8 0.02 view 60 -30
#dump_modify 2 pad 3

#dump 3 all movie 25 movie.mpg type type &
axes yes 0.8 0.02 view 60 -30
#dump_modify 3 pad 3

thermo 50
run 250

--:--- in.melt All L1 (Fundamental)-----

Wrote /home/user/.emacs

VMDで可視化する(5)

- 編集した in.melt の実行

```
$ lammmps < in.melt
```

エラーメッセージなしで

Total wall time が表示されれば
正常終了

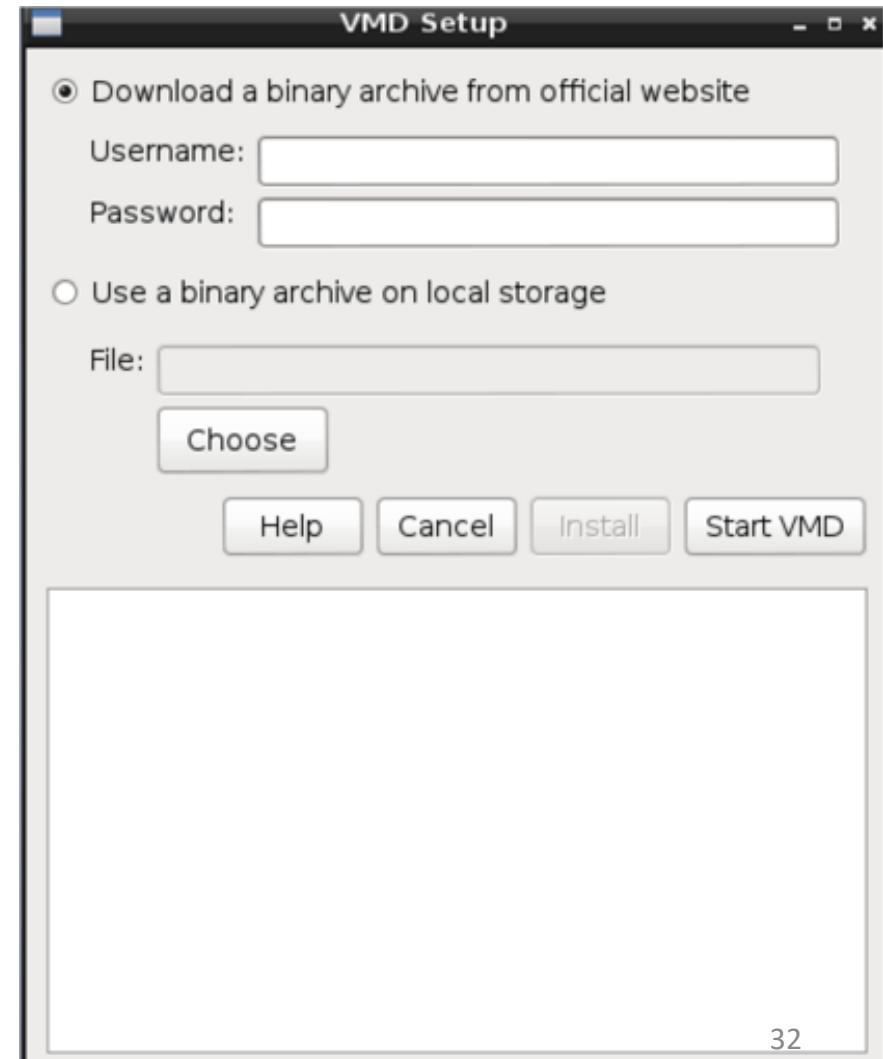
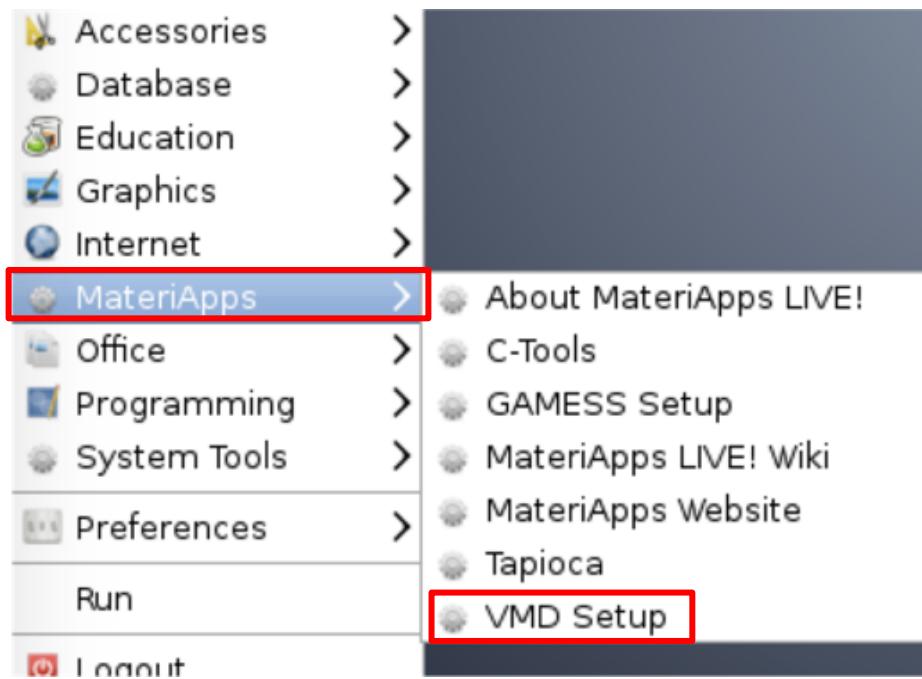
- dump.melt というファイルができているか確認する。

```
$ ls
```

dump.melt	in.melt.org	log.5Oct16.melt.g++.4
in.melt	log.5Oct16.melt.g++.1	log.lammmps

VMDで可視化する(6)

- VMDの起動
「MateriApps」の「VMD Setup」を選択



VMDで可視化する(7)

- VMDのユーザ登録
(すでにユーザIDとパスワードをお持ちの方は必要ありません。)
- VMDのウェブサイトへアクセスする。
「Download VMD」でgoogle検索する。
Download VMD - Theoretical and Computational Biophysics Group
<https://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=VMD>
のサイトへアクセス

NIH CENTER FOR MACROMOLECULAR MODELING & BIOMINFORMATICS | UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Type Keywords SEARCH

THEORETICAL and COMPUTATIONAL BIOPHYSICS GROUP



Home Research Publications Software Instruction News Galleries Facilities About Us

Software Downloads

Download VMD:

VMD is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. Visit the [VMD website](#) for complete information and documentation.

Selecting an archive below will lead to a user registration and login page. Your download will continue after you have registered or logged in.

Version 1.9.4 LATEST ALPHA (2017-12-21) Platforms:

Latest pre-release ALPHA test version

クリック

- LINUX_64 OpenGL, CUDA, OptiX, OSPRay (Linux (RHEL 6.7 and later) 64-bit Intel/AMD x86_64 SSE, with CUDA 9.x, OptiX, OSPRay)
- MacOS X OpenGL (32-bit Intel x86) (Apple Mac OS X (10.10.x or later) with hardware OpenGL (native bundle))

VMDで可視化する(8)

- username と password を入力
(新しく自分で決める。)

Registration/Login

You will need a username and password to download software.

If this is your first download, please choose a username and password to register.
Current NAMD or VMD users, please enter your existing username and password.

Username:

① Username と Passwordを入力
(注)rist2018は使えません

Password:

② ボタンを押す

[Continue with registration or download](#)

Your download will continue after you have registered or logged in.

すでにユーザIDとパスワードをお持ちの方は、この作業は必要ありません。

VMDで可視化する(9)

- ユーザ登録をする

New User Registration

New User Registration for 'rist2018':

すでにユーザIDとパスワードをお持ちの方は、この作業は必要ありません。

First and Last Name:

Kanako Yoshizawa

Email Address:

yoshizawa@rist.or.jp

① ご自身の情報を入力

Affiliation:

Academic Government Industrial Other (specify)

The number of people using TCBG software at my site is:

1 2-4 5-10 11-20 21 or more

I use TCBG software primarily for:

Research Teaching Commerce Personal

The work I do with TCBG software is funded (at least partially) by NIH:

Yes No

Re-enter password for confirmation:

Register

② ボタンを押す

VMDで可視化する(10)

- 承諾をする

Software Downloads

Welcome! Account created for 'rist2018'.

Please remember your password for future downloads.

You may avoid logins for 6 months by saving a cookie on your browser:

VMD 1.9.4 for Linux (RHEL 6.x) 64-bit x86_64 w/ SSE, CUD

To download this software you must agree to abide by the terms of the following license:

UNIVERSITY OF ILLINOIS
VISUAL MOLECULAR DYNAMICS SOFTWARE LICENSE AGREEMENT

Upon execution of this Agreement by the party identified below ("Licensee"),
The Board of Trustees of the University of Illinois ("Illinois"), on behalf
of The Theoretical and Computational Biophysics Group ("TCBG") in the
Beckman Institute, will provide the Visual Molecular Dynamics ("VMD") software in
Executable Code and/or Source Code form ("Software") to Licensee, subject to the
following

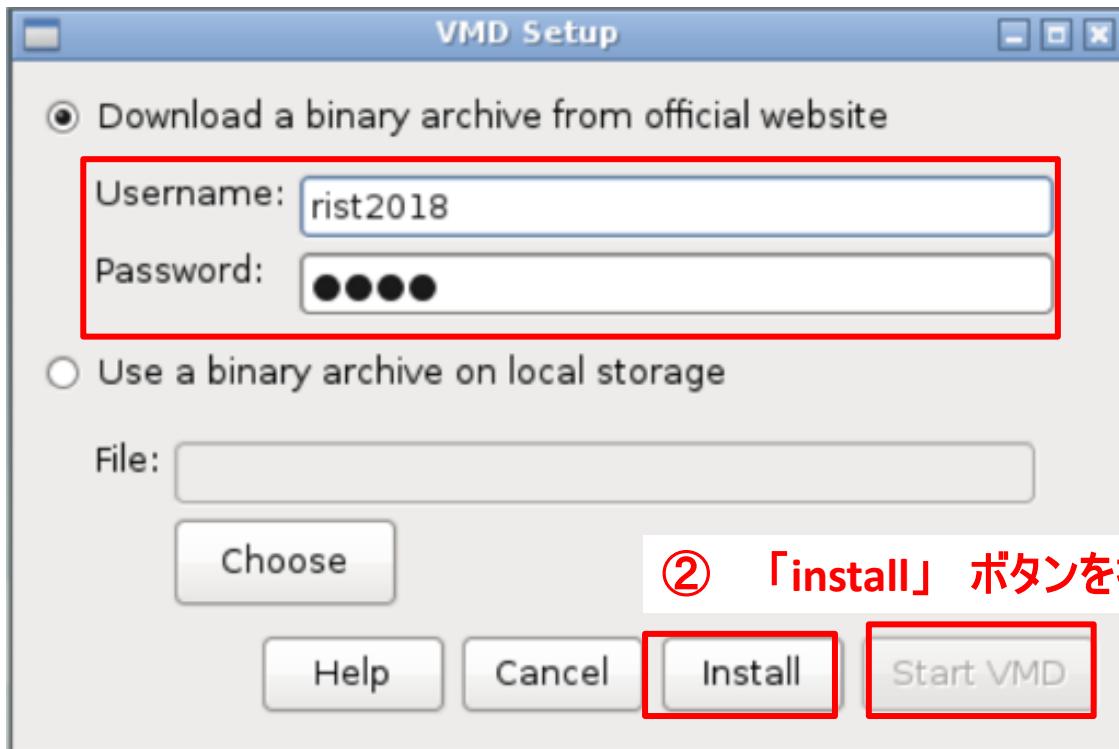
terms and conditions. For purposes of this Agreement, Executable Code is the
compiled code, which is ready to run on Licensee's computer. Source code
consists of a set of files which contain the actual program commands that are
compiled to form the Executable Cod

ボタンを押す

すでにユーザIDとパス
ワードをお持ちの方は、
この作業は必要ありま
せん。

VMDで可視化する(11)

- 「VMD Setup」において、username と password を入力



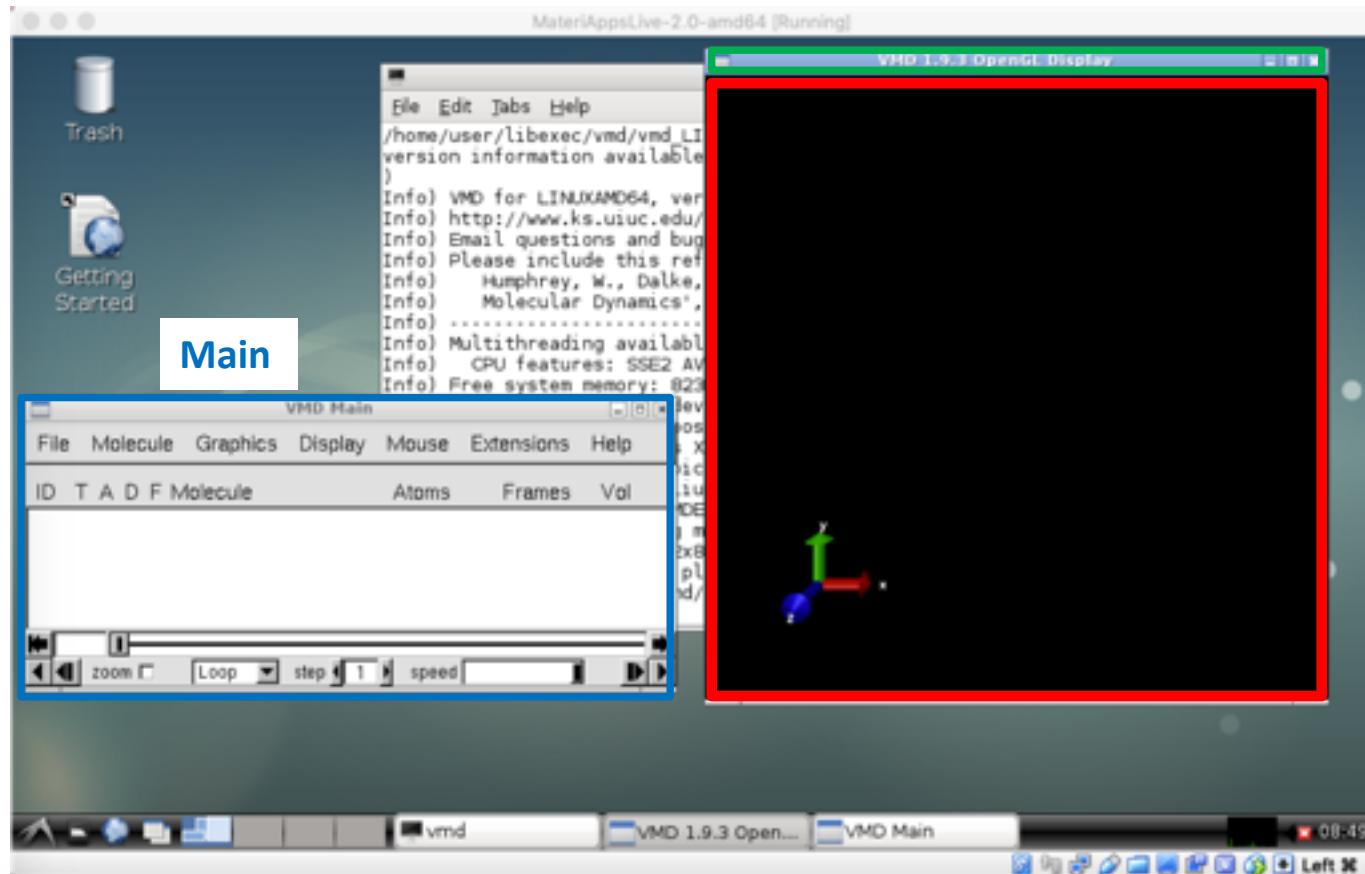
① username と
password を入力

② 「install」 ボタンを押す

③ setup が終わったら
「Start VMD」 ボタンを押す

VMDで可視化する(12)

- VMDのウインドウ

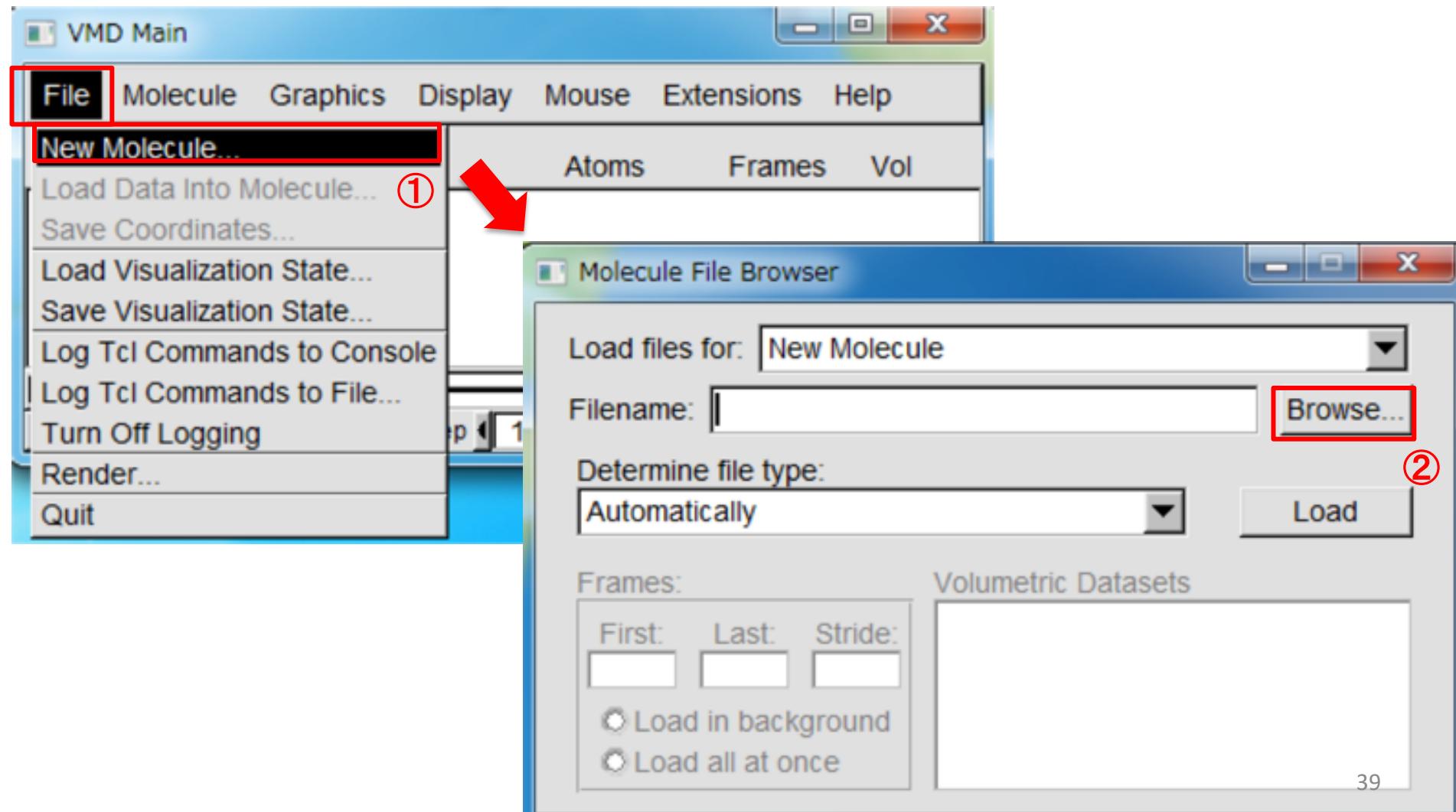


(注) 緑の部分が隠れてしまい、ウインドウが移動できないときは、

- Mac の場合
option を押しながら ドラッグ
- Windows の場合 ALT を押しながら ドラッグ
すると、赤で囲まれたウインドウを移動できる。

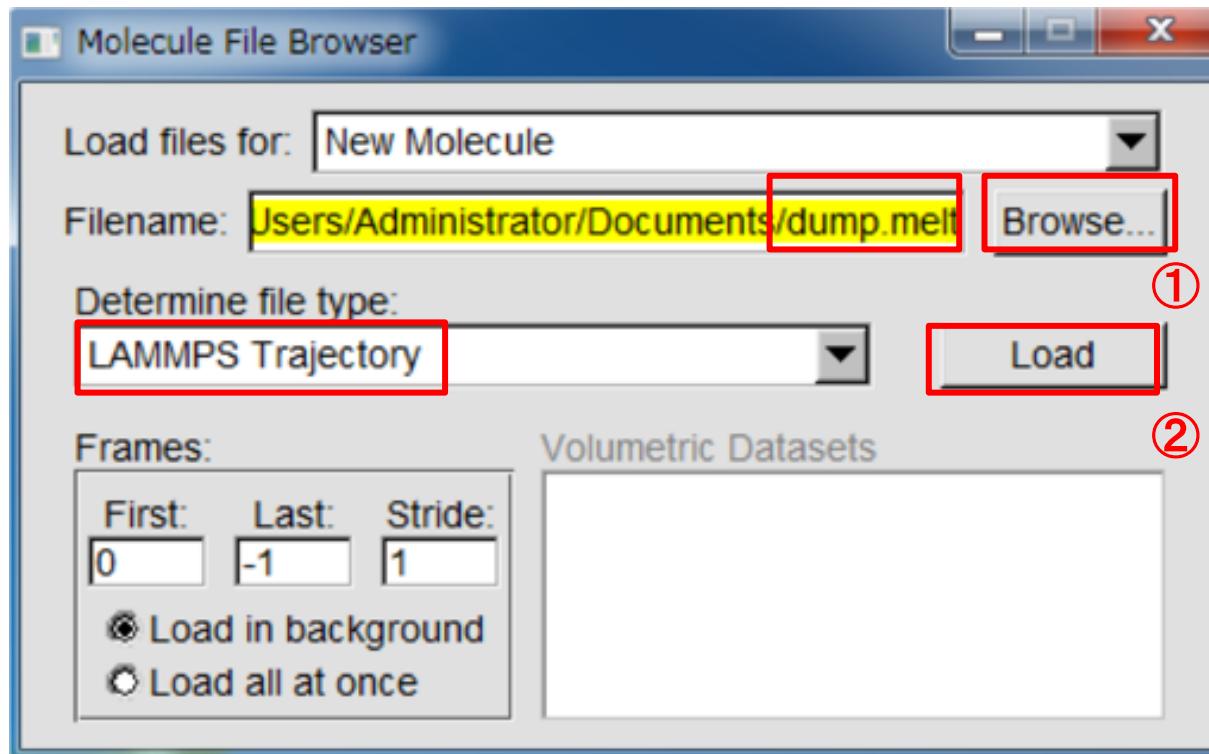
VMDで可視化する(13)

- dump.melt を開く



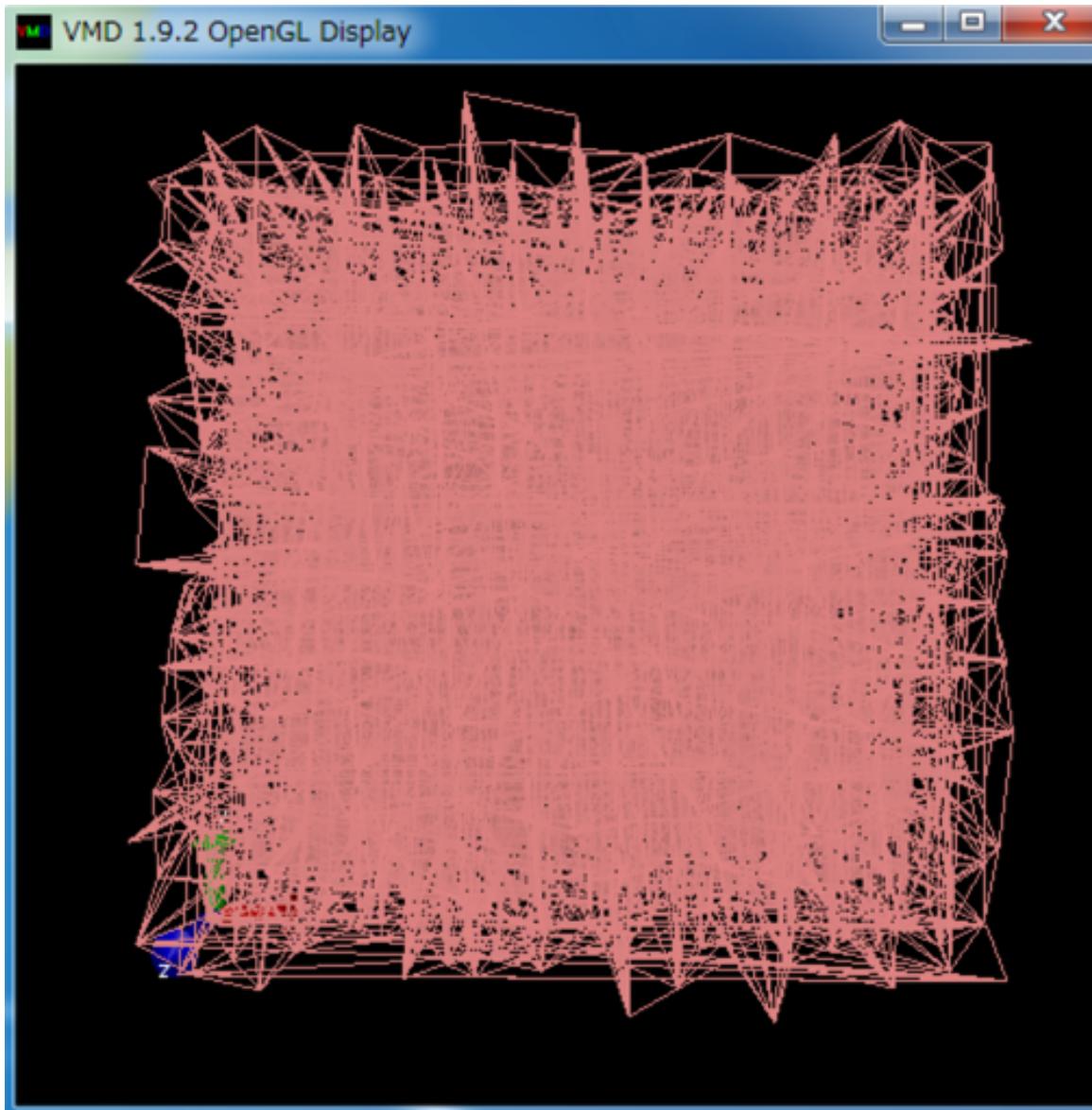
VMDで可視化する(14)

- ファイルタイプの選択 → 「LAMMPS Trajectory」



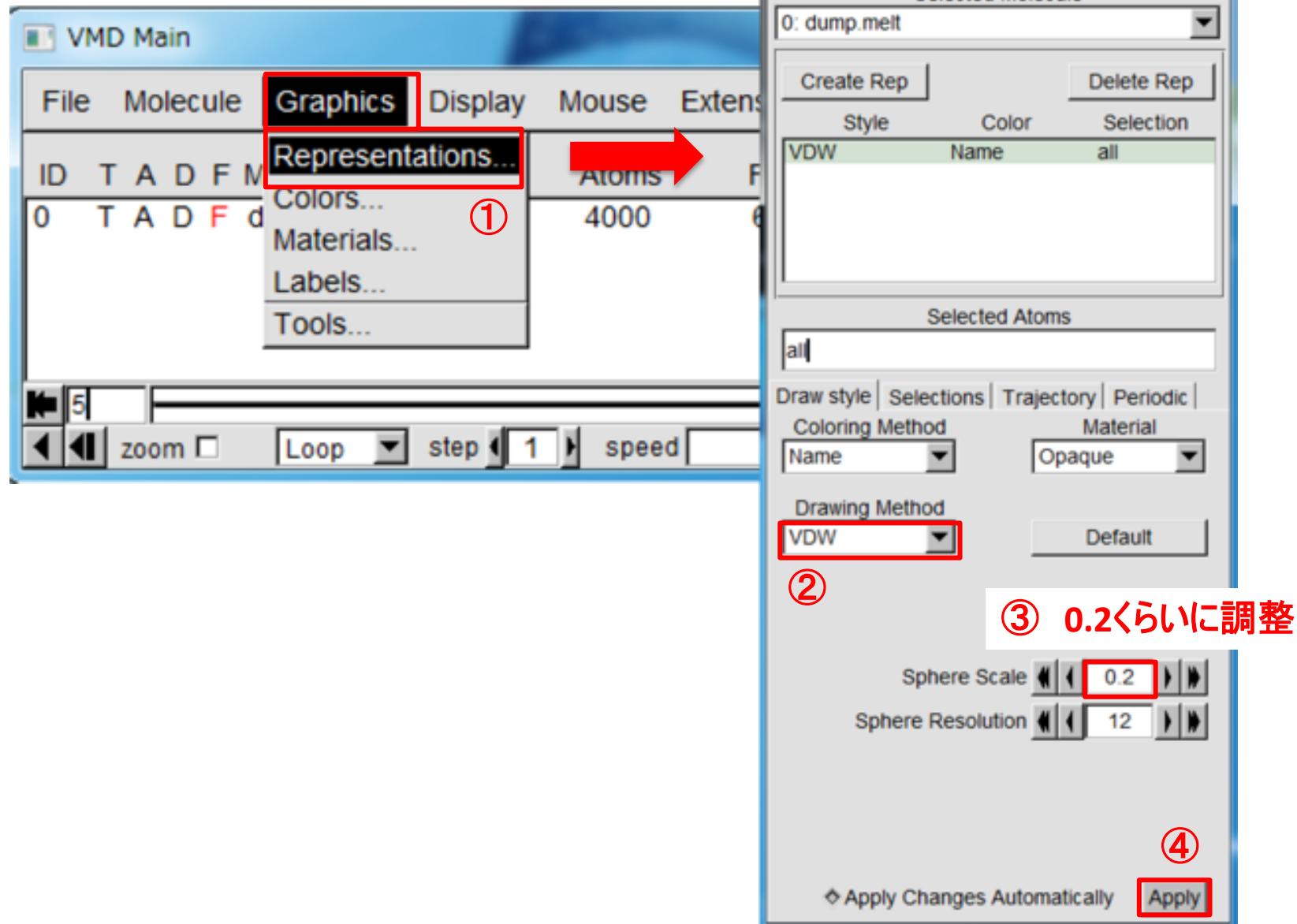
VMDで可視化する(15)

- ディスプレイの起動(Linesの表示)



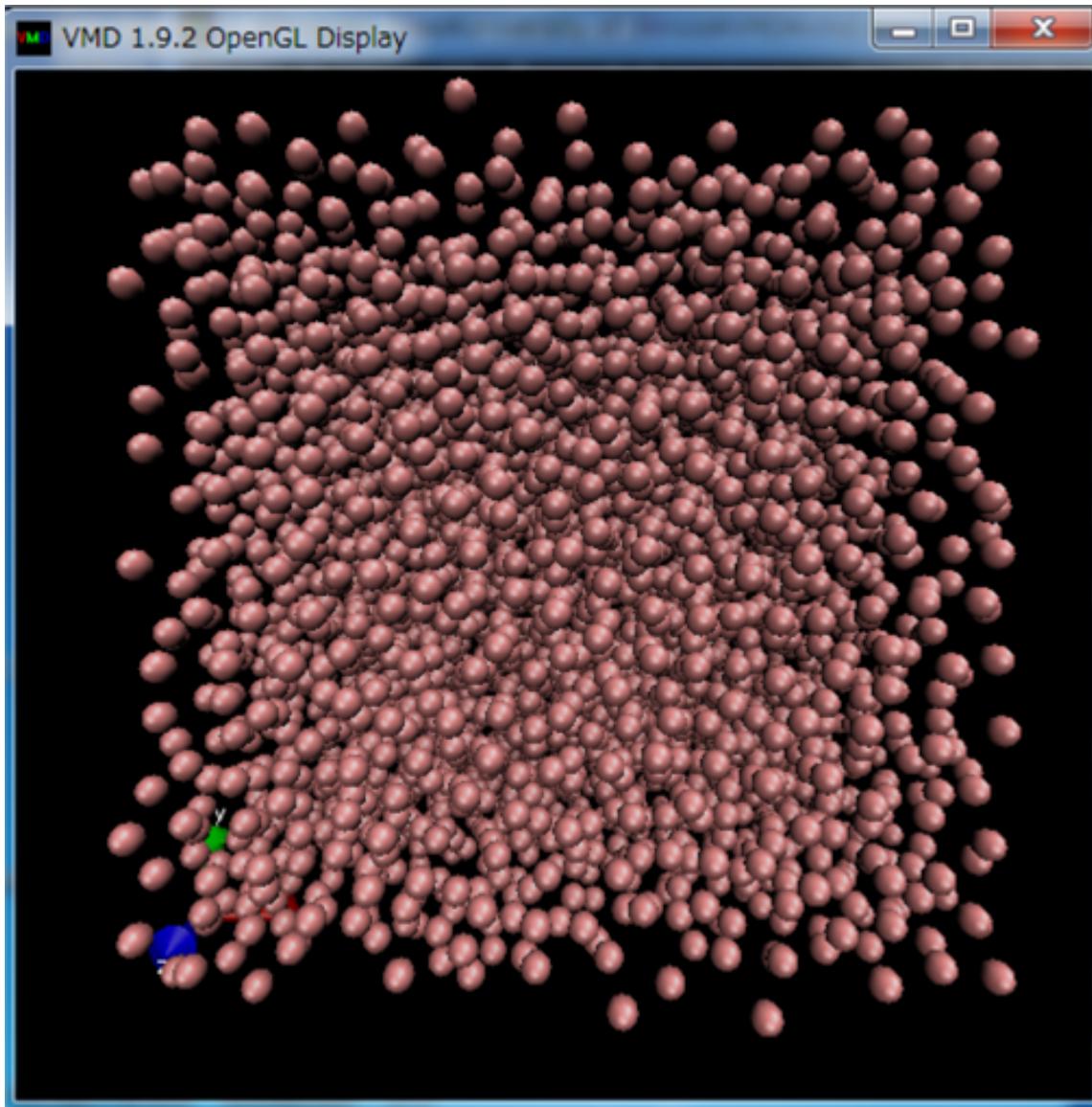
VMDで可視化する(16)

- 表示形式の調整



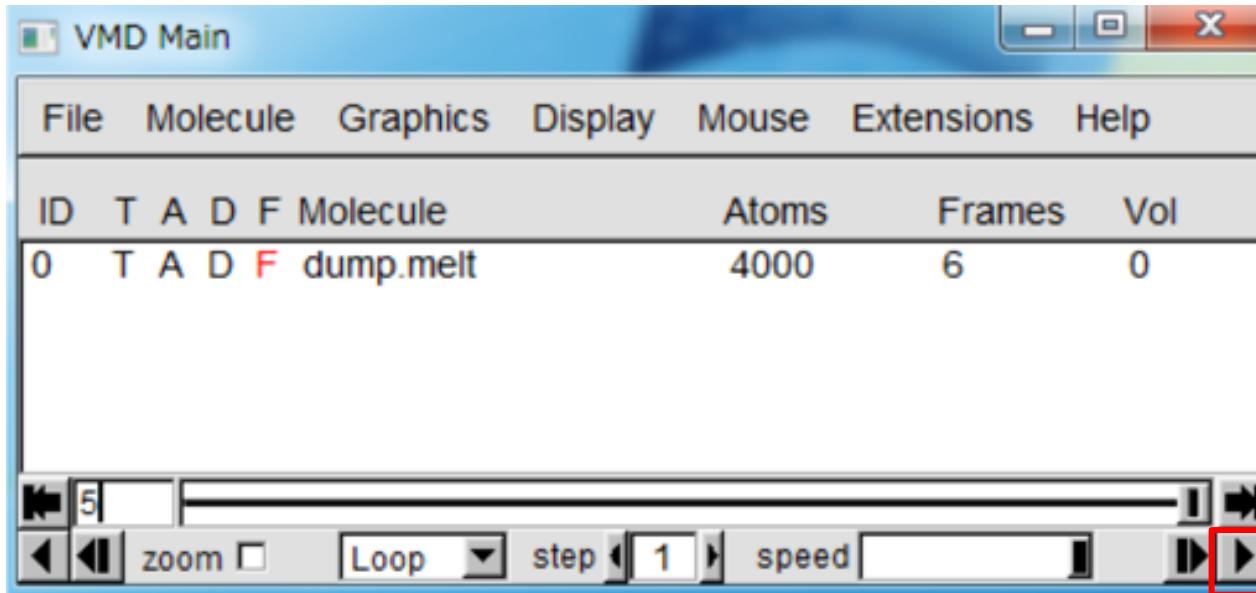
VMDで可視化する(17)

- ディスプレイ(VDWの表示)



VMDで可視化する(18)

- MDスタート



ここをクリック

micelle と colloid の可視化の課題

1. micelle と colloid に対しても dumpファイルを出力
2. micelle と colloid の dumpファイルをVMDで可視化

micelle の可視化(1)

```
# 2d micelle simulation

dimension 2      2次元の計算

neighbor 0.3 bin
neigh_modify delay 5

atom_style bond

# Soft potential push-off

read_data data.micelle      データファイル data.micelle を読み込む
special_bonds fene

...
fix 1 all nve      NVEアンサンブル

...
thermo 1000

dump 1 all atom 2000 dump.micelle ← # 取る
...
...
```

micelle の可視化(2)

- 編集した in.micelle の実行

```
$ lammps < in.micelle
```

エラーメッセージなしで
Total wall time が表示されれば
正常終了

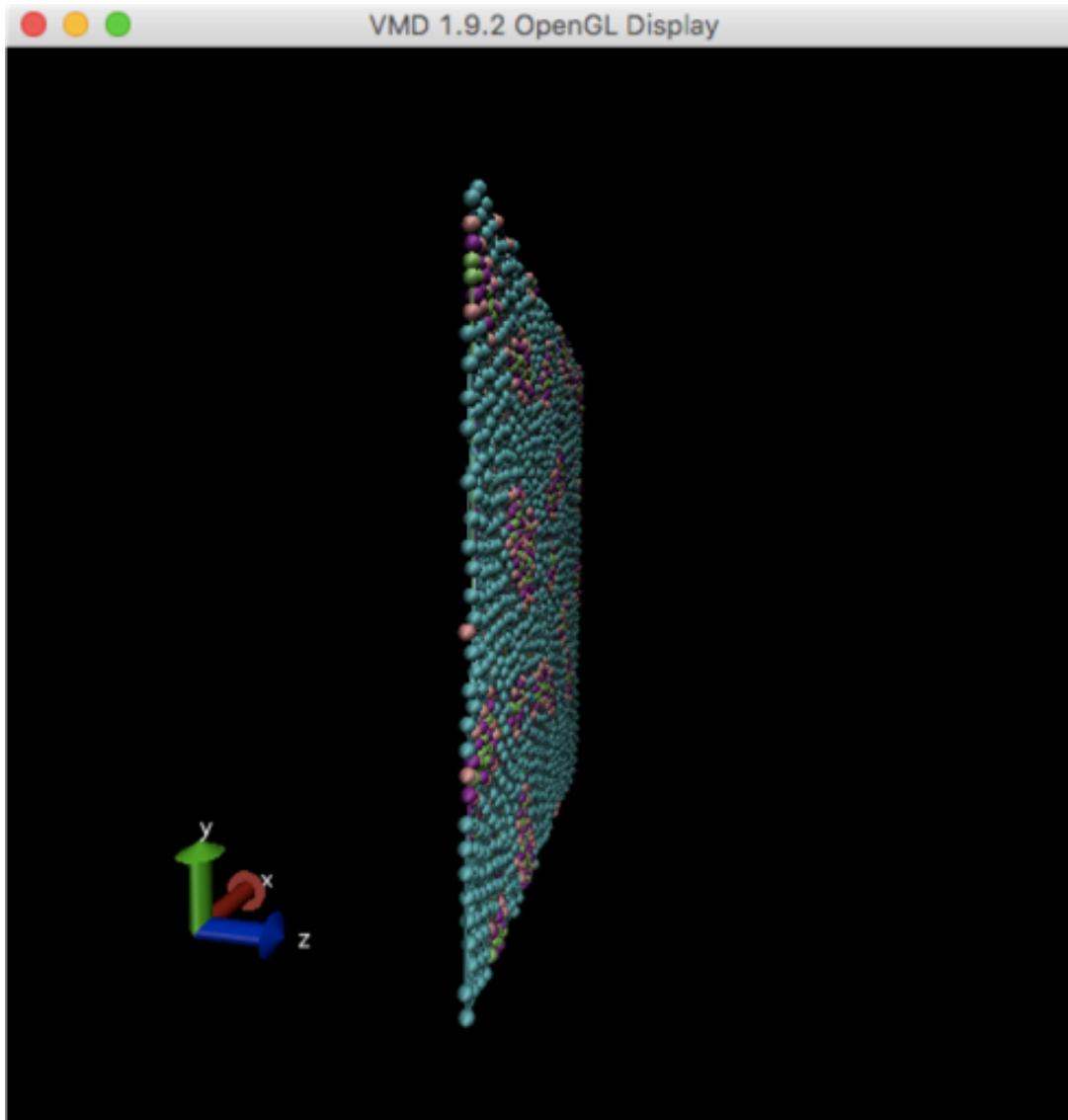
- dump.micelle というファイルができているか確認する。

```
$ ls
```

```
data.micelle  dump.micelle  log.5Oct16.micelle.g++.1.gz  log.lammps  
def.micelle  in.micelle  log.5Oct16.micelle.g++.4.gz
```

micelle の可視化(3)

- dump.micelle を開く



可視化をすると
計算している系が
わかりやすい。

micelle と colloid の3次元化の課題

1. 2次元の micelle と colloid が、3次元になるように入力ファイルを編集する
 2. これらの結果をVMDで可視化する
-
- 作業ディレクトリを作る。(例: \$HOME/lammps/micelle3d など)
\$ cd \$HOME/lammps
\$ mkdir micelle3d
\$ mkdir colloid3d
 - micelle(colloid)ディレクトリから in.micell(in.colloid) をコピーする
\$ cp ./micelle/in.micelle ./micelle3d
\$ cp ./micelle/data.micelle ./micelle3d
\$ cp ./colloid/in.colloid ./colloid3d

micelleの3D化について(1)

1. 変更点(inファイル)

- Dimension文コメントアウト
- Replicateコマンド (read_dataの下)
replicate 1 1 36
- NPT計算に変更

fix 1 all npt temp 1.0 1.0 1.0 iso 3.0 3.0 10.0

(計算時間がかかるようなら、run 60000 の回数を減らす)

2. 変更点(dataファイル)

- PBC-boxの大きさ

0.0000000E+00 0.996023889 zlo zhi

ここで、 $0.996023889 = 35.85686 / 36$

micelleの3D化について(2)

in.micelle(入力ファイル)

```
# 3d micelle simulation
```

```
bond_coeff 1 50.0 0.75
```

```
# dimension 2
```

2次元の設定の
コメントアウト
→ 3次元

```
velocity all create 0.45 2349852
```

```
neighbor 0.3 bin
```

```
variable prefactor equal ramp(1.0,20.0)
```

```
neigh_modify delay 5
```

```
atom_style bond
```

```
# Soft potential push-off
```

```
read_data data.micelle
```

data.micelle
のデータファイルが
読み込まれる

```
replicate 1 1 36
```

z方向に36個
複製する

```
special_bonds fene
```

```
pair_style soft 1.12246
```

```
pair_coeff * * 0.0 1.12246
```

```
bond_style harmonic
```

```
fix 1 all npt temp 1.0 1.0 1.0 iso 3.0 3.0 10.0
```

```
fix 2 all temp/rescale 100 0.45 0.45 0.02 1.0
```

```
fix 3 all adapt 1 pair soft a * * v_prefactor
```

```
# fix 4 all enforce2d
```

コメントアウト

```
thermo 50
```

```
run 1000
```

```
unfix 3
```

1(=ID)という名称で、
全て(=all)の粒子に対して、NPT積分を行う。温
度(=temp)は、計算開始時は1.0、終了時は
1.0、時間単位1.0で
dampする。
isoは、外圧を指定する。

→ 続く

micelleの3D化について(3)

in.micelle(入力ファイル)

→ 続く

Main run

pair_style lj/cut 2.5

solvent/head - full-size and long-range

pair_coeff 1 1 1.0 1.0 2.5

pair_coeff 2 2 1.0 1.0 2.5

pair_coeff 1 2 1.0 1.0 2.5

tail/tail - size-averaged and long-range

pair_coeff 3 3 1.0 0.75 2.5

pair_coeff 4 4 1.0 0.50 2.5

pair_coeff 3 4 1.0 0.67 2.5

solvent/tail - full-size and repulsive

pair_coeff 1 3 1.0 1.0 1.12246

pair_coeff 1 4 1.0 1.0 1.12246

thermo 1000

dump 1 all atom 2000 dump.micelle

#dump 2 all image 2000 image.*.jpg type type
zoom 1.6

#dump_modify 2 pad 5 adiam 1 0.5 adiam 2 1.5
adiam 3 1.0 adiam 4 0.75

#dump 3 all movie 2000 movie.mpg type type
zoom 1.6

#dump_modify 3 pad 5 adiam 1 0.5 adiam 2 1.5
adiam 3 1.0 adiam 4 0.75

reset_timestep 0

run 60000

計算時間がかかるようなら
減らす

micelleの3D化について(5)

data.micelle(データファイル)

LAMMPS 3d micelle data file

1200 atoms

1 1.000000

2 1.000000

3 1.000000

4 1.000000

0 angles

0 dihedrals

0 impropers

Atoms

4 atom types

1 139 2 0.000 0.000 0.000

1 bond types

2 0 1 1.195 0.000 0.000

0 angle types

3 0 1 2.390 0.000 0.000

0 dihedral types

4 0 1 3.586 0.000 0.000

0 improper types

5 0 1 4.781 0.000 0.000

...

0.0000000E+00 35.85686 xlo xhi

0.0000000E+00 35.85686 ylo yhi

0.0000000E+00 0.996023889 zlo zhi

データファイルのフォーマット

データファイル

LAMMPS Description

5 atoms

4 bonds

1 atom types

1 bond types

-10.0 10.0 xlo xhi

-10.0 10.0 ylo yhi

-10.0 10.0 zlo zhi

Masses

1 1

Atoms

1 1 1 0.0 0.0 0.0

2 1 1 0.0 0.0 1.0

3 1 1 0.0 0.0 2.0

4 1 1 0.0 0.0 3.0

5 1 1 0.0 0.0 4.0

Bonds

1 1 1 2

2 1 2 3

3 1 3 4

4 1 4 5

データフォーマット

Masses

atom type番号, 質量

周期境界条件
のwrap数

Atoms

粒子番号, Mol-ID, atom type, x, y, z, **wx**, **wy**, **wz**

Velocities

粒子番号, vx, vy, vz

Bonds

ボンド番号, bond type, bond1, bond2

※ 速度データを含む場合
(AtomsとBondsの間)

Velocities

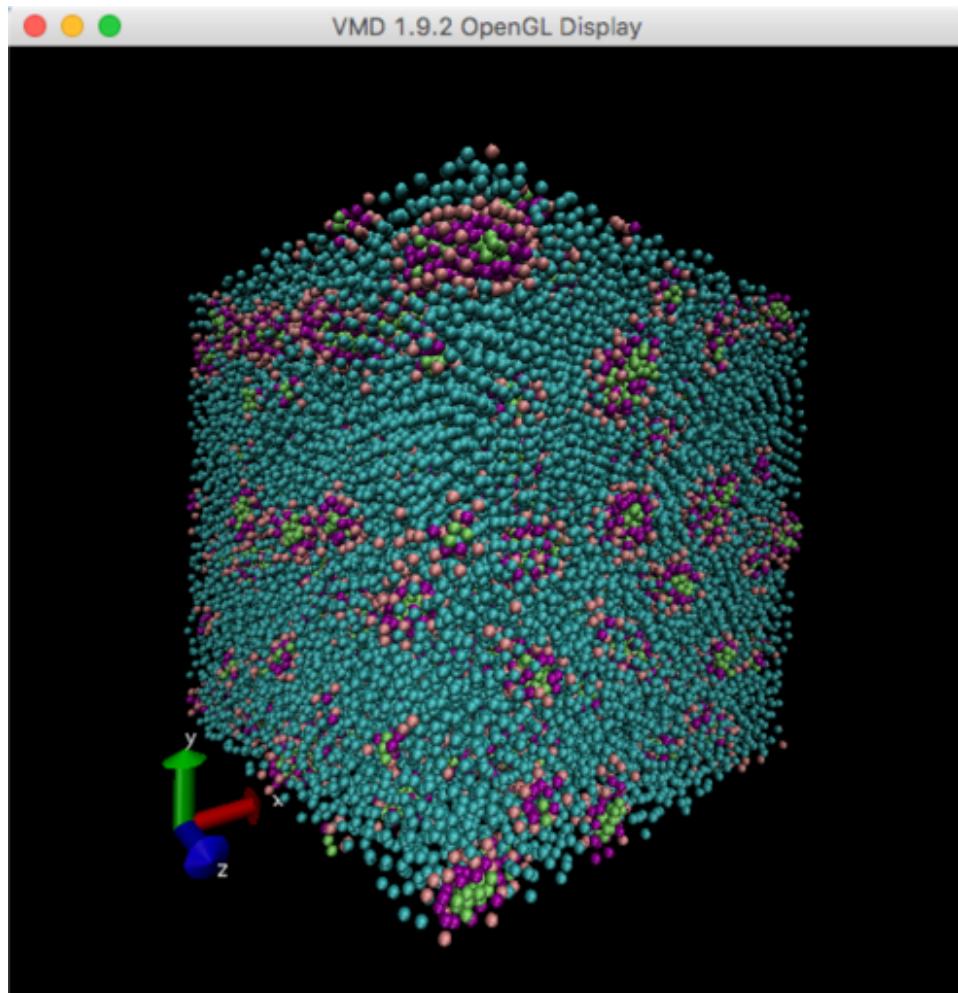
1 0.5 0.5 0.5

※ MassesとAtomsの間に、力場 パラメータを記録する場合もある。

※ Bondsの後に、Angles, Dihedrals, Impropersを記録する場合もある。

micelleの3D化について(4)

- in.micelle と data.micelle の編集後、実行
\$ lammmps < in.micelle
- dump.micelle を開く



VMDで可視化すると、3Dになったことが確認出来る

colloidの3D化について(1)

1. 変更点(inファイル)

- Dimension文コメントアウト
- Latticeコマンド → lattice sc 0.001
濃度を下げる、オーバラップを解除
- Regionコマンドを変更。
region box block 0 30 0 30 0 30
- Fix-enforce2dを、コメントアウト

(計算時間がかかるようなら、run 50000 の回数を減らす)

colloidの3D化について(2)

in.colloid(入力ファイル)

```
# Big colloid particles and small LJ particles
units      lj
atom_style sphere
# dimension 2
lattice    sc 0.001
region     box block 0 30 0 30 0 30
create_box 2 box
create_atoms 1 box

set        group all type/fraction 2 0.96 23984
set        type 1 mass 9
set        type 2 mass 1
velocity   all create 1.44 87287 loop geom
# multi neighbor and comm for efficiency

neighbor   1 multi
neigh_modify delay 0
comm_modify mode multi

# colloid potential

pair_style colloid 12.5
pair_coeff 1 1 1.0 1.0 5.0 5.0 12.5
pair_coeff 1 2 5.0 1.0 5.0 0.0 7.0
pair_coeff 2 2 10.0 1.0 0.0 0.0 2.5

fix        1 all npt temp 2.0 2.0 1.0 iso 0.0 1.0 10.0
drag 1.0 &
          mtk no pchain 0 tchain 1
# fix      2 all enforce2d

dump       1 all atom 1000 dump.colloid
```

→ 続く

colloidの3D化について(3)

in.colloid(入力ファイル)

→ 続く

```
#dump      2 all image 1000 image.*.jpg type  
type &  
#       zoom 1.5 center d 0.5 0.5 0.5  
#dump_modify  2 pad 5 adiam 1 5.0 adiam 2 1.5  
  
#dump      3 all movie 1000 movie.mpg type type  
&  
#       zoom 1.5 center d 0.5 0.5 0.5  
#dump_modify  3 pad 5 adiam 1 5.0 adiam 2 1.5  
  
thermo_style custom step temp epair etotal  
press vol  
thermo      1000  
  
timestep    0.005
```

run 50000

計算時間がかかるようなら
減らす

サイズを大きくした計算(1)

- 作業ディレクトリを作る。(例: \$HOME/lammps/melt2)
\$ cd \$HOME/lammps
\$ mkdir melt2
- 作成した作業ディレクトリへ移動する。(\$HOME/lammps/melt2)
\$ cd melt2
- meltディレクトリから in.melt をコピーする
\$ cp ../melt/in.melt ./
- melt の入力ファイルを開く
\$ emacs in.melt

サイズを大きくした計算(2)

```
# 3d Lennard-Jones melt

units          lj
atom_style     atomic

lattice        fcc 0.8442
region         box block 0 40 0 40 0 40      256,000原子になる
create_box     1 box
create_atoms   1 box
mass           1 1.0

velocity       all create 3.0 87287

pair_style     lj/cut 2.5
pair_coeff     1 1 1.0 1.0 2.5

neighbor       0.3 bin
neigh_modify   every 20 delay 0 check no

fix            1 all nve
```

サイズを大きくした計算(3)

- 編集した in.melt の実行

```
$ lammps < in.melt
```

- ログファイルの確認

```
$ cat log.lammps
```

...

Created 256000 atoms

...

サイズを大きくした計算(4)

examples/melt

実行時間(時:分:秒)

iMac:
2.8 GHz Intel Core i7

	手持ちのPC(iMac)	
プロセス数	1	4
4000原子1,000ステップ	0:00:02	0:00:01
4000原子10,000ステップ	0:00:29	0:00:11
256,000原子1,000ステップ	0:03:06	0:01:13
256,000原子10,000ステップ	0:31:15	0:12:48
2,048,000原子1,000ステップ	0:27:04	0:10:06
2,048,000原子10,000ステップ		1:46:38

フラットMPI

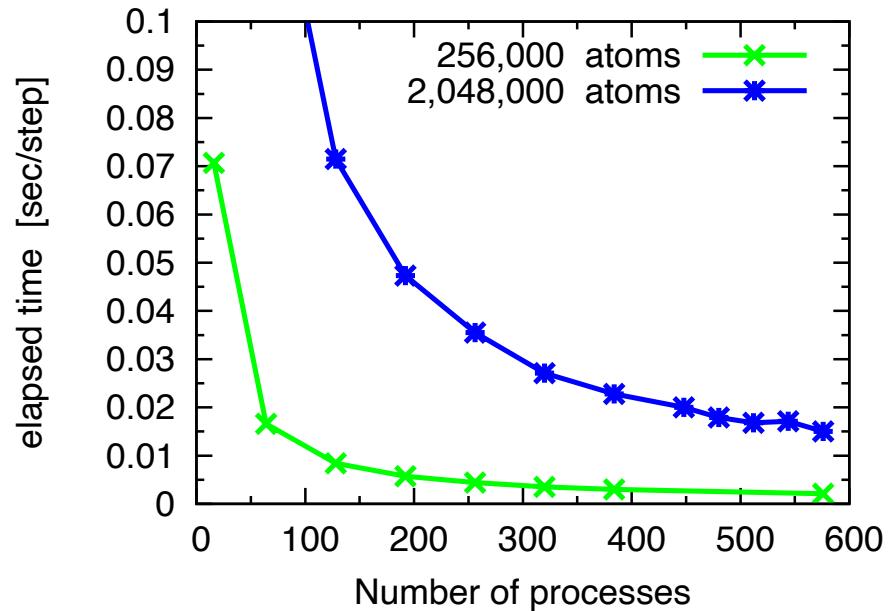
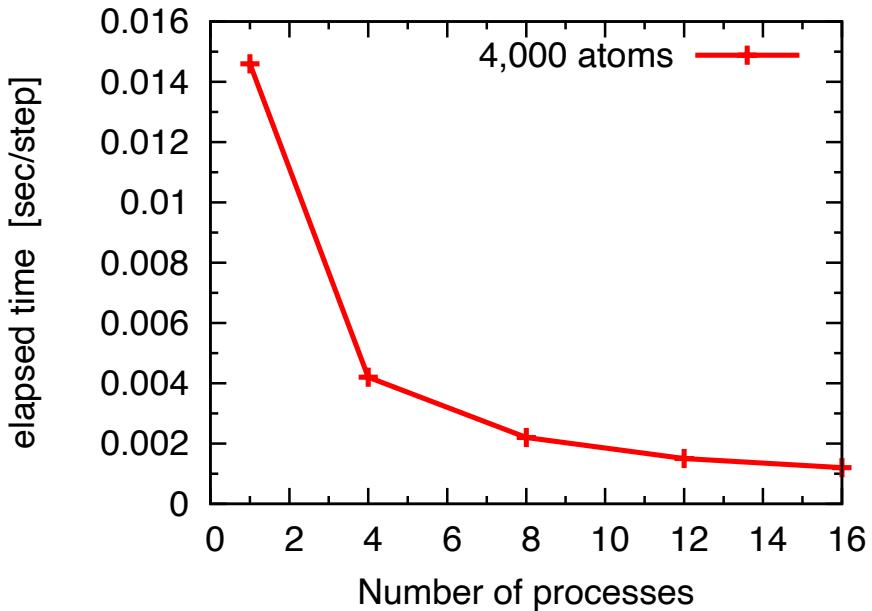
FX10		
16	256	576
0:00:01	0:00:05	
0:00:12	0:00:03	
0:01:11	0:00:04	0:00:02
0:11:47	0:00:44	0:00:21
0:08:56	0:00:36	0:00:15
1:28:35	0:05:55	0:02:30

- 主にプロセス数当たりの粒子数減少に伴う実行時間の減少
→ 計算規模(原子数など)が小さくて、プロセス数を増やすと、通信等により実行時間が増加する場合がある。
- 計算規模が大きくなる → HPCの利用へ

サイズを大きくした計算(5)

examples/melt

フラットMPI(原子数:4,000、256,000、2,048,000)



4,000原子:～16プロセス

256,000原子:～600プロセス

2,048,000原子:600プロセス以上



性能測定はRISTの高度化支援へ

restartを行う(1)

コマンドの紹介

- write_restart/restart: 計算途中ファイル出力する。
- read_restart: 計算を再開する。
- clear: リセット

このコマンドを使用すると、1つの入力ファイルで複数のジョブを順番に実行できる。

(例)

(commands for 1st simulation)

clear

(commands for 2nd simulation)

- log: LAMMPSのログファイル(log.lammps)をクローズして、指定されたログファイルにログを取る。

restartを行う(2)

- examples の hugoniostat を実行する

exmapleファイルのコピー(hugoniostat)

```
$ cp -r /usr/share/lammps/examples/hugoniostat ./
```

(注)in.hugoniostatの実行には、SHOCKパッケージが必要。

SHOCKパッケージがないと、

ERROR: Unknown fix style nphug (./modify.cpp:854)

Last command: fix myhug all nphug temp 1.0 1.0 10.0 z 40.0 40.0 70.0 drag 0.0
tchain 1 pchain 0

のエラーが出る。

- hugoniostatディレクトリの中を確認する

```
$ ls
```

in.hugoniostat log.5Oct16.hugoniostat.g++.1

log.27Nov18.hugoniostat.g++.1 log.5Oct16.hugoniostat.g++.4

log.27Nov18.hugoniostat.g++.4

restartを行う(3)

in. hugoniostat(入力ファイル)

```

# This script reproduces stress trajectories from Fig. 1 in
# Ravelo, Holian, Germann, and Lomdahl, PRB 70 014103 (2004)
...
units          lj
boundary       p p p
atom_style     atomic
...
#
# Define initial velocity
velocity       all create 0.01 87287 mom yes rot yes dist gaussian
write_restart  restart.equil      restartファイルの出力

# Start Run #1
clear          clear によってリセットを行い、
               read_restart で、保存しておいた restart.equil の構造から計算を再開する
read_restart   restart.equil      restartファイルの入力
...

```

restartを行う(4)

- in.hugoniostat の実行

```
$ lammps < in.hugoniostat > log.hugoniostat
```

(log.hugoniostat には、標準出力の結果が保存される。)

- 出力されているログファイルを確認する。

```
$ ls
```

```
in.hugoniostat      log.5Oct16.hugoniostat.g++.1 log.lammps
```

```
log.27Nov18.hugoniostat.g++.1 log.5Oct16.hugoniostat.g++.4 restart.equil
```

```
log.27Nov18.hugoniostat.g++.4 log.hugoniostat      stress_vs_t.dat
```

- log.hugoniostat には、Run1～3の結果がまとめて出力されている。

【実習】

- Run 1～3ごとのログファイルを出力する

restartを行う(5)

in. hugoniostat(入力ファイル)の編集

```
...  
# Start Run #1  
  
log log.nodrag  
  
clear  
read_restart restart.equil  
  
...  
  
# Start Run #2  
  
log log.drag  
  
clear  
read_restart restart.equil  
  
...  
  
# Start Run #3  
  
log log.nhchains  
  
clear  
read_restart restart.equil  
  
...
```

赤文字の部分を追記すると、

保存しておいた restart.equil
(write_restart restart.equilで出力した)の
構造を
Run1～3の計算条件で計算とき、
ログファイルとして
Run1: log.nodrag
Run2: log.drag
Run3: log.nhchains
にそれぞれ保存

restartを行う(6)

- 編集した in.hugoniostat の実行
\$ lammmps < in.hugoniostat > log.hugoniostat
- log.nodrag, log.drag, log.nhchains というファイルができているか確認する。
\$ ls
in.hugoniostat log.5Oct16.hugoniostat.g++.1 log.hugoniostat
log.nodrag
log.27Nov18.hugoniostat.g++.1 log.5Oct16.hugoniostat.g++.4 log.lammmps
restart.equil
log.27Nov18.hugoniostat.g++.4 **log.drag** **log.nhchains**
stress_vs_t.dat
- それぞれのログファイルの中身を確認する。
\$ cat log.nodrag
など
- log.hugoniostat には、Run1～3(log.nodrag, log.drag, log.nhchains) の結果が output されている。

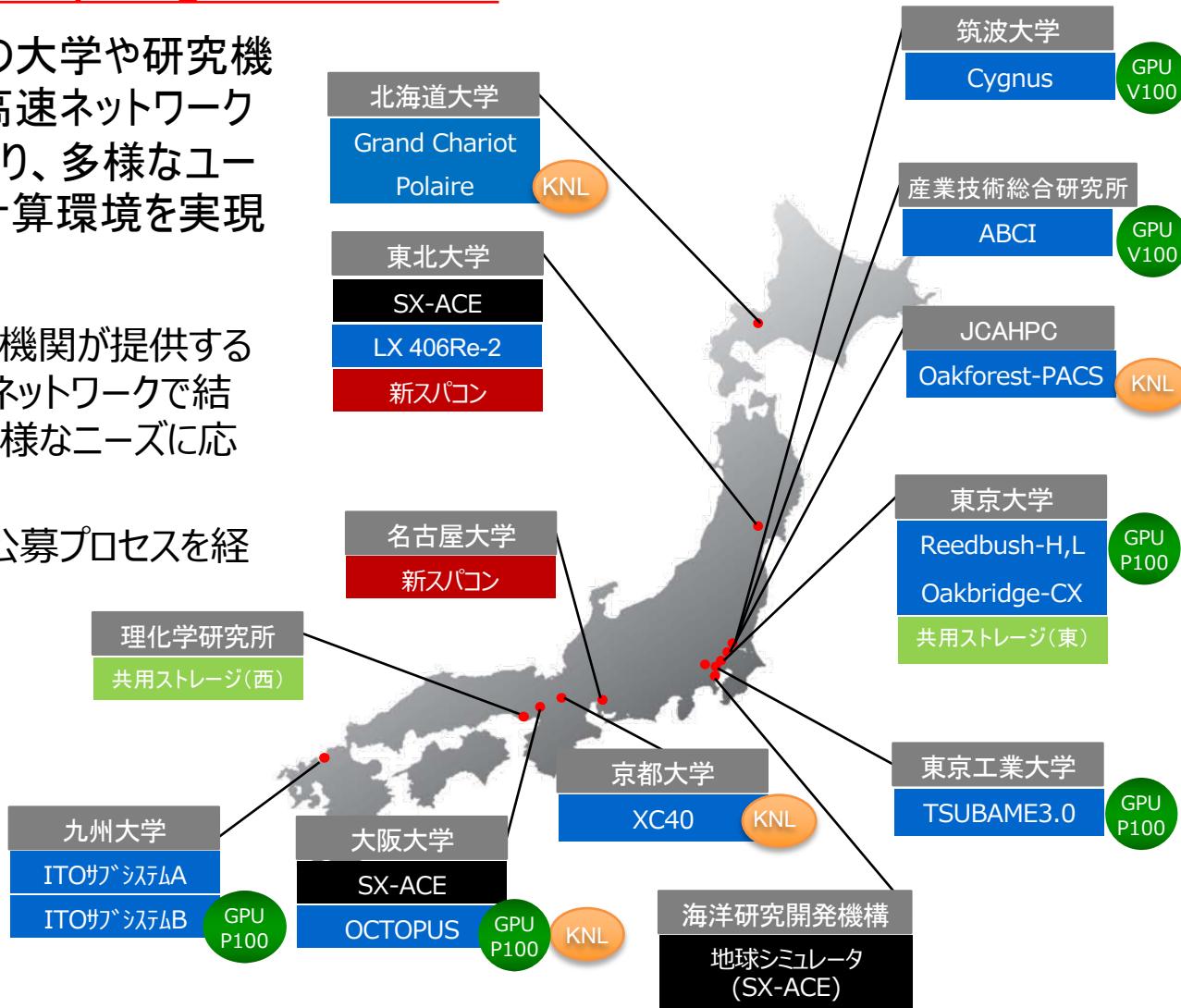
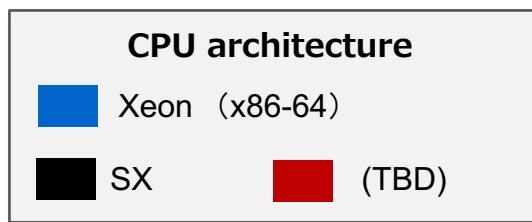
HPCIにプリインストールされた アプリ情報

HPCI計算資源(令和元年9月5日版)

HPCI: High Performance Computing Infrastructure

HPCIシステムは、「京」と全国の大学や研究機関の計算資源やストレージを高速ネットワーク(SINET5)で結んだシステムであり、多様なユーザーニーズに応える革新的な計算環境を実現しています。

- フラグシップ計算機と大学・研究機関が提供する計算機資源・ストレージを高速ネットワークで結び、利用者支援を充実させ、多様なニーズに応える革新的な計算環境を実現
- アカデミアのみでなく産業界も、公募プロセスを経てこの計算環境を利用可能。



ネットワーク基盤：学術情報ネットワーク (SINET5) を利用

HPCIの情報



★ HPCIポータルサイト

<http://www.hpci-office.jp>

The header of the HPCI website features the acronym 'HPCI' in large letters, followed by 'High Performance Computing Infrastructure'. A green button labeled 'English' is located at the top center. To the right are search fields for 'サイト内検索' (Site Search) and '詳細検索' (Advanced Search), along with a magnifying glass icon. On the far right, there's a '文字サイズ' (Text Size) button with three font icons. The main navigation menu includes links for 'HPCIについて' (About HPCI), '利用案内・申請' (Usage Instructions/Application), '利用支援' (Support), 'HPCI研究成果' (HPCI Research Results), 'イベント・講習会' (Events/Seminars), '広報' (Public Relations), and '利用者向け情報' (Information for Users).

High Performance Computing Infrastructure

English

サイト内検索

詳細検索

文字サイズ : A A A

HPCIについて 利用案内・申請 利用支援 HPCI研究成果 イベント・講習会 広報 利用者向け情報

提供する主な内容

- HPCIについて
 - 利用案内・申請
 - 課題の申請
 - 課題選定の結果
 - 利用支援
 - HPCI研究成果
 - イベント・講習会
 - 広報
 - 利用者向け情報
 - 各種手続きについて
 - プリインストールされたアプリケーションソフトウェア

HPCIポータルサイト:ソフトウェア検索システム

http://www.hpci-office.jp/pages/project_categories_hardware



▼ HPCIシステムで利用可能なソフトウェアの検索

この条件で検索する >

検索範囲
過去に利用可能であったソフトウェアと計算機資源の情報を含める

ソフトウェア名
検索条件:
※すべてを含む(AND) いずれかを含む(OR)

ソフトウェア分類
すべての分類
シミュレーション
分子動力学 量子化学 物理物理 計算生物学 流体解析 構造/計算解析
電磁界解析 マルチフィジックス 粒子系 原子/気体 地図学
データサイエンス
機械学習 統計解析/データ分析 バイオインフォマティクス 分散データ処理
ブリボスト
汎用可視化ソフトウェア メッシュ操作
ライブラリ等
MPI通信 数値計算 形式処理 画像処理 データ圧縮 データ形式 Python関連
その他のライブラリ等
開発環境
コンパイラ/インタプリタ 強制分析 デバッガ その他の開発環境
システム基盤
OS ジョブ管理 クラウド基盤

- 検索機能

- ソフトウェア名による検索 [NEW]
- ソフトウェアの分類を選択してソフトウェア情報を絞り込み [NEW]
- 過去に利用可能であったソフトウェアと計算機資源の情報を含めた検索 [NEW]
- 特定の資源提供機関及び計算機資源を選択したソフトウェア情報の絞り込み

- ソフトウェアに関する情報を追加
 - ソフトウェアの概要説明 [NEW]
 - 計算機資源毎のインストール情報や利用条件等
 - 利用課題の多いソフトウェアを中心に関連情報を案内 [NEW]
 - 各ソフトウェアを利用した課題の利用報告書へのリンク [NEW]
 - 各資源提供機関のソフトウェア関連ページへのリンク [NEW]

HPCIIにプリインストールされている国プロアプリ

	北大	東北大	筑波大	JCAHPC	東大	東工大	名大	京大	阪大	九大
	Grand Chariot	LX 406Re-2	Cygnus	Oakforest-PACS	Oakbridge-CX	TSUBAME3.0	CX400	XC40	OCTOPUS	ITO-A/B
ABINIT-MP	●			●	●	●	●	●	●	●
FrontFlow/blue	●			●	●	●	●	●	●	●
FrontISTR	●			●	●	●	●	●	●	●
GENESIS	●		● (G)	●	●	● (+G)	●	●	● (+G)	● (+G)
HΦ	●	●		●	●	●	●	●	●	●
MODYLAS	●	●		●	●	●	●	●	●	●
NTChem	●	●		●	●	●	●	●	●	●
OpenMX	●	●		●	●	●	●	●	●	●
PHASE/0	●			●	●	●	●	●	●	●
SALMON	●	●		●	●	●	●	●	●	●
SMASH	●	●		●	●	●	●	●	●	●

【凡例】黒:整備済、赤:2019年度整備予定、緑:2019年度バージョンアップ予定、
(G):GPU版のみ、(+G):CPU版に加えてGPU版を整備

(2019年10月現在)

付録

材料系シミュレーションの情報収集

物質科学シミュレーションのポータルサイト

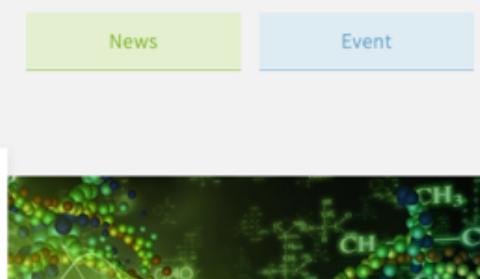
MateriApps <https://ma.issp.u-tokyo.ac.jp/>




The screenshot shows the MateriApps homepage. At the top, there's a search bar with placeholder text "MateriAppsについて 気になるワードを入力" and a QR code. Below the search bar, there are links for "JP / EN", "お問合せ / アプリ開発依頼", and "265件". The main navigation menu includes "NEWS / 講習会・イベント", "アプリ一覧", "アプリ詳細検索", "キーワード解説", "レビュー", "事例", and "アプリコンシェルジュ". A banner at the top says "インストールしないでアプリを試せる「MateriApps LIVE!」". The "Category" section features a large green button labeled "Category" and several application cards. One card for "FermiSurfer for Android" by "32aki" (@kawamitsukii) shows screenshots of the app interface.

News / Event

NEWS / 講習会・イベント



The screenshot shows the "News / Event" section. It has two tabs: "News" (green background) and "Event" (blue background). An event card for "CCMSハンズオン: DCore講習会 (TIA“かけはし”連携講座)" on "2019 10/4" is displayed. The card includes a small thumbnail image of a molecular simulation visualization.

- 200の物質科学アプリケーションやツールを紹介
- 「やりたいこと」からアプリケーションを検索
検索タグ:「特徴」「対象」「手法・アルゴリズム」など
- 開発者の声を利用者に届ける
アプリ紹介、開発者ページ、アプリの魅力・将来性・応用性
- フォーラム(掲示板)を利用した意見交換
- 講習会情報・web講習会・更新情報
- アクセス数
2018年度:2116 ユーザ/月、11267 PV/月
2019上期(4~9月):4380 ユーザ/月、16534 PV/月
- アプリコンシェルジュ
利用者から計算物質科学やアプリに関する質問・要望にMateriApps開発チームが解答
- レビュー

(1) Windowsでの利用(バイナリが配布されている)

- LAMMPS-ICMS Windows Installer Repository

<http://rpm.lammps.org/windows.html>

例) バイナリファイル名(32ビット): **lammps-32bit-latest.exe**

(注) 64ビット版のバイナリもある

Cygwin(<https://www.cygwin.com/>)を用いて実行

(MPIを用いるときは、<http://www.mpich.org/static/tarballs/1.4.1p1/mpich2-1.4.1p1-win-ia32.msi> をインストール)

実行例(個人の環境によって異なる可能性あり)

- **lammps-32bit-latest.exe** をダウンロードして実行(インストール)
- Cygwin を起動
サンプルのあるディレクトリ(デフォルトのままインストールした場合)
</cygdrive/c/Program%20Files/LAMMPS%20-bit%20160124/Examples>
実行ファイルのあるディレクトリ
</cygdrive/c/Program%20Files/LAMMPS%20-bit%20160124/bin>
\$ **mp_serial.exe <(input file)** ← シリアルで実行
\$ **mp_mpi.exe <(input file)** ← MPIで実行

(2) Linuxでの利用

● ソースをコンパイル

実行例(個人の環境によって異なる可能性あり)

ターミナルを起動

ソース(**lammps-stable.tar.gz**)を解凍

→ ディレクトリ **lammps-11Aug17** を作る (11Aug17版の場合)

```
$ tar zxvf lammps-stable.tar.gz
```

必要に(利用したい機能に)応じてパッケージの追加

http://lammps.sandia.gov/doc/Section_packages.html

```
$ cd lammps-11Aug17/src
```

```
$ make package-status ← パッケージの一覧の表示
```

```
$ make yes-colloid : (例) COLLOIDパッケージを追加
```

- シリアルで実行

```
$ cd ./lammps-7Dec15/src
```

```
$ make serial
```

```
$ ./lmp_serial < (input file)
```

- MPIで実行(**OpenMPIなどが必要**)

```
$ cd ./lammps-7Dec15/src
```

```
$ make mpi
```

```
$ ./lmp_mpi < (input file)
```

(3) Macでの利用(ソースをコンパイル:コンパイラが必要)

OS X El Capitan での実行例(個人の環境によって異なる可能性あり)

ターミナルを起動

ソース(lammps-stable.tar.gz)を解凍 → ディレクトリ lammps-11Aug17 を作る

- シリアルでコンパイル

```
$ cd lammps-11Aug17/src  
$ make serial  
$ lmp_serial < (input file)
```

- MPIでコンパイル(OpenMPIなどが必要)

```
$ cd lammps-11Aug17/src  
$ make mpi  
$ lmp_mpi < (input file)
```

(注)Mac用のmakefile: lammps-11Aug17/src/MAKE/MACHINES/Makefile.mac
もある。

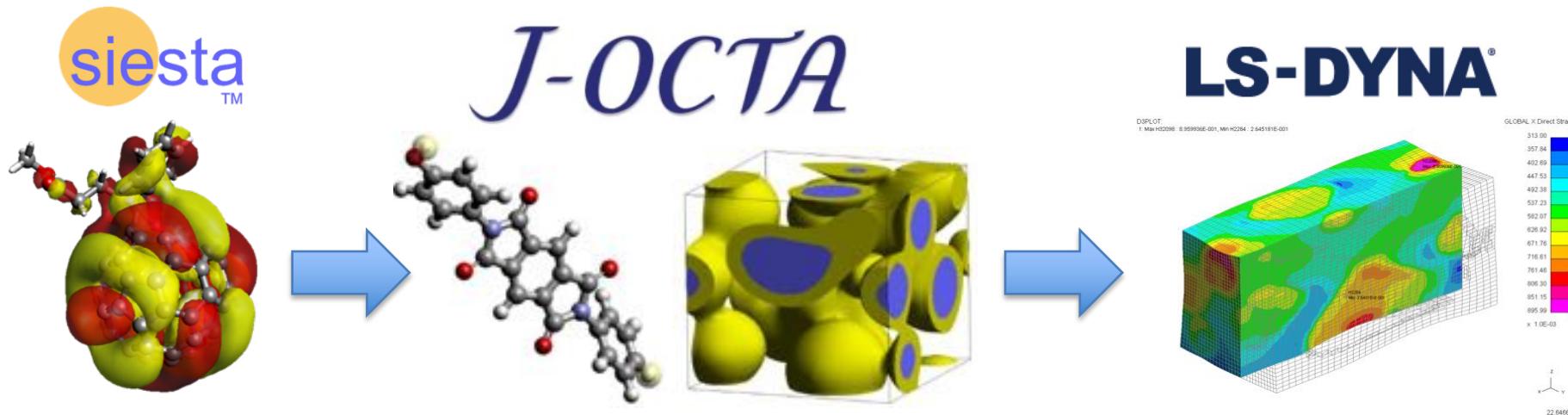
LAMMPSのGUIアプリ(1)

- Winmostar <https://winmostar.com/>
 - 株式会社クロスアビリティが開発・販売
 - 企業ユースで15万円より(トライアル版あり)
 - サポート、講習会(関東・関西)、個別カスタマイズあり
 - 平均20ダウンロード/日
 - 国内外にシングルライセンス775ユーザ、機関ライセンス48契約
 - 古典MD:LAMMPS, Gromacs, Amber
 - 第一原理:QuantumESPRESSO, Gaussian, GAMESS, NWChem, MOPAC
 - 材料系国プロアプリ(OpenMX, SMASH , DC-DFTB, MODYLAS, Fragment ER)にも対応



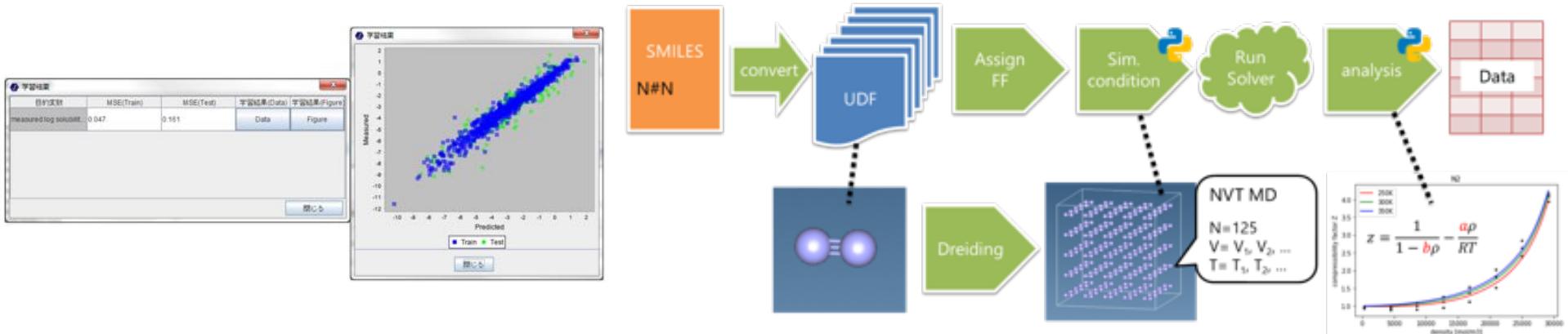
LAMMPSのGUIアプリ(2)

- J-OCTA <https://cae.jsol.co.jp/product/material/jocta/>
 - 2005年 JSOL(日本総研)が開発、OCTAの商用版
※ Student Edition(機能制限付きの無償版)がある
 - 第一原理計算から非線形有限要素法まで(SIESTA、LS-DYNAとI/F)
 - 分子動力学では、LAMMPS、GROMACS、HOOMDをマルチスケールシミュレーション統合環境に拡張でき、多機能なプリポストが利用可能
 - ✓ COGNACのデータに基づいて、各ソルバーのデータと相互に変換
 - ✓ 各ソルバーの入力ファイルが自動生成され、手動修正も可能
 - ✓ 結果ファイルは自動的にCOGNACデータに変換される



LAMMPSのGUIアプリ(3)

- J-OCTA <https://cae.jsol.co.jp/product/material/jocta/>
 - 機械学習(深層学習)を用いたQSPR(定量的構造物性相関)機能
 - 分子構造(SMILES)と物性データの学習を、GUI上で簡単に実施。
 - いくつかの物性の学習済みモデルは、JSOLから継続提供される。
 - 学習用データをシミュレーションで構築するための分子モデリングAPI機能
 - 力場や電荷の取得がPythonスクリプト内から可能(Windows、Linux)。
 - 多数の物質に対して、分子構造(SMILES)をもとにMDのバルク構造作成、物性計算をバッチ実行。



LAMMPSのGUIアプリ(3)

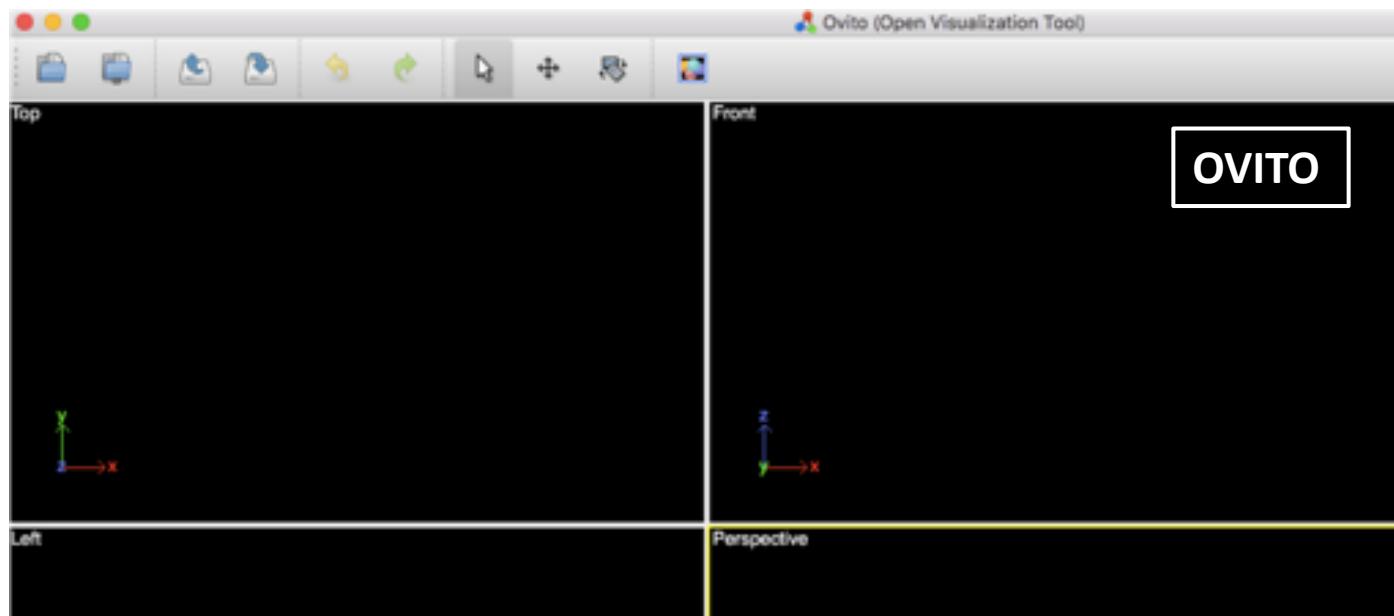
- その他のツール

Pre/Post Processing Tools for use with LAMMPS(紹介サイト)

<http://lammps.sandia.gov/prepost.html>

可視化ツール

- VMD <http://www.ks.uiuc.edu/Research/vmd>
- OVITO <http://www.ovito.org>



LAMMPSのベンチマークについて(1)

- LAMMPSのベンチマーク

<http://lammps.sandia.gov/bench.html>

ホームページで、ベンチマークの結果と、ベンチマーク用の
入力ファイルの公開

(cf. Gromacs: http://www.gromacs.org/About_Gromacs/Benchmarks)

Machine specifications

These are the parallel [machines](#) for which benchmark data is given for the CPU benchmarks below. See the Kokkos, Intel, and GPU sections for machine specifications for those GPU and Phi platforms.

The "Processors" column is the most number of processors on that machine that LAMMPS was run on. Message passing bandwidth and latency is in units of Mb/sec and microsecs at the MPI level, i.e. what a program like LAMMPS sees. More information on machine characteristics, including their "birth" year, is given [at the bottom of the page](#).

Vendor/Machine	Processors	Site	CPU	Interconnect	Bandwidth	Latency
Dell T7500 dual hex-core desktop	12	SNL	3.47 GHz Xeon	on-chip	??	??
Xeon/Myrinet cluster	512	SNL	3.4 GHz dual Xeons (64-bit)	Myrinet	230	9
IBM p690+	512	Daresbury	1.7 GHz Power4+	custom	1450	6
IBM BG/L	65536	LLNL	700 MHz PowerPC 440	custom	150	3
Cray XT3	10000	SNL	2.0 GHz Opteron	Cstar	1100	7
Cray XT5	1920	SNL	2.4 GHz Opteron	Cstar	1100	7

→自身の環境でのベンチマークと比較する

LAMMPSのベンチマークについて(2)

- 力場による性能の違い

<http://lammps.sandia.gov/bench.html#potentials>

Interatomic potential comparisons

The following table summarizes the CPU cost of various potentials, as implemented in LAMMPS, each for a system commonly modeled by that potential. The desktop machine these were run on is described below. The last 3 entries are for [VASP](#) timings, to give a comparison with DFT calculations. The details for the VASP runs are described below.

The listed timing is CPU seconds per timestep per atom for a one processor (core) run. Note that this is per timestep, as is the ratio to LJ; the timestep size is listed in the table. In each case a short 100-step run of a roughly 32000 atom system was performed. The speed-up is for a 4-processor run of the same 32000-atom system. Speed-ups greater than 4x are due to cache effects.

To first order, the CPU and memory cost for simulations with all these potentials scales linearly with the number of atoms N, and inversely with the number of processors P when running in parallel. This assumes the density doesn't change so that the neighbors per atom stays constant as you change N. This holds for N/P ratios larger than some threshold, say 1000 atoms per processor. Thus you can use this data to estimate the run-time of different size problems on varying numbers of processors.

Potential	System	# Atoms	Timestep	Neighs/atom	Memory	CPU	LJ Ratio	P=4 Speed-up	Input script	Tarball
Granular	chute flow	32000	0.0001 tau	7.2	33 Mb	2.08e-7	0.26x	4.28x	in.granular	bench_granular.tar.gz
FENE bead/spring	polymer melt	32000	0.012 tau	9.7	8.4 Mb	2.86e-7	0.36x	3.78x	in.fene	bench_fene.tar.gz
Lennard-Jones	LJ liquid	32000	0.005 tau	76.9	12 Mb	8.01e-7	1.0x	3.56x	in.lj	bench_lj.tar.gz
DPD	pure solvent	32000	0.04 tau	41.3	9.4 Mb	1.22e-6	1.53x	3.54x	in.dpd	bench_dpd.tar.gz
EAM	bulk Cu	32000	5 fmsec	75.5	13 Mb	1.87e-6	2.34x	3.83x	in.eam	bench_eam.tar.gz
REBO	polyethylene	32640	0.5 fmsec	149	33 Mb	3.18e-6	3.97x	3.61x	in.rebo	bench_rebo.tar.gz

≈

→ ホームページで公開されている情報を活用する

(注)LAMMPSの開発グループが公開している情報

<http://lammps.sandia.gov/bench.html>

ホームページで、公開されている入力ファイルについて

(LAMMPSのパッケージ lammps-11Aug17/bench/POTENTIALS
にも同じファイルが置いてある。)

– 書式が古い → 最新版のLAMMPSで動かすには、変更が必要

(例)

“Interatomic potential comparisons”

<http://lammps.sandia.gov/bench.html#potentials>

にある

in.granular LAMMPS (30 Jun 2012)版、ならそのまま動くが、
最新版では、

communicate → comm_modify mode

などの変更の必要

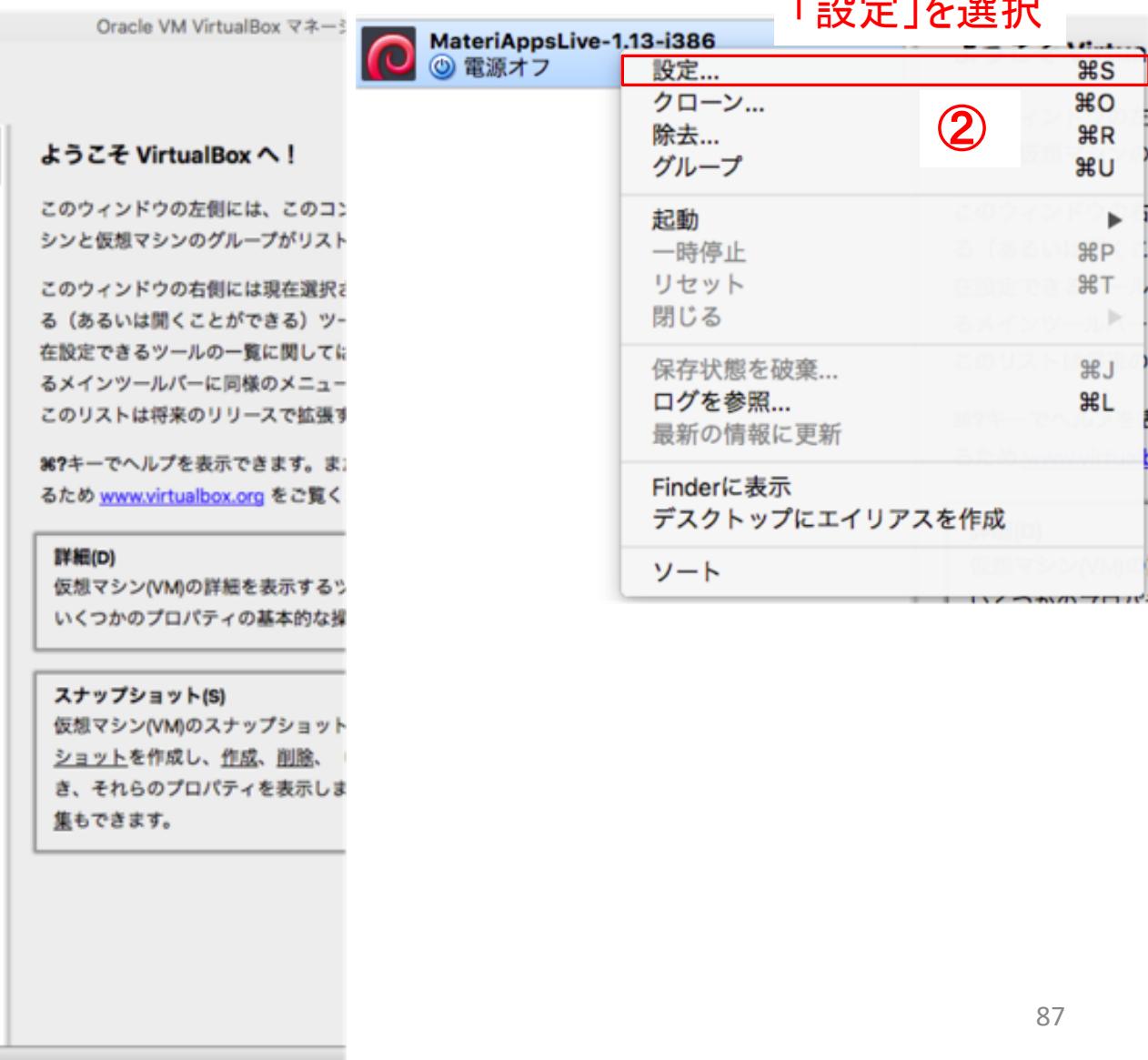
仮想マシンからUSBへのアクセス設定(1)

(1) VirtualBox を起動する



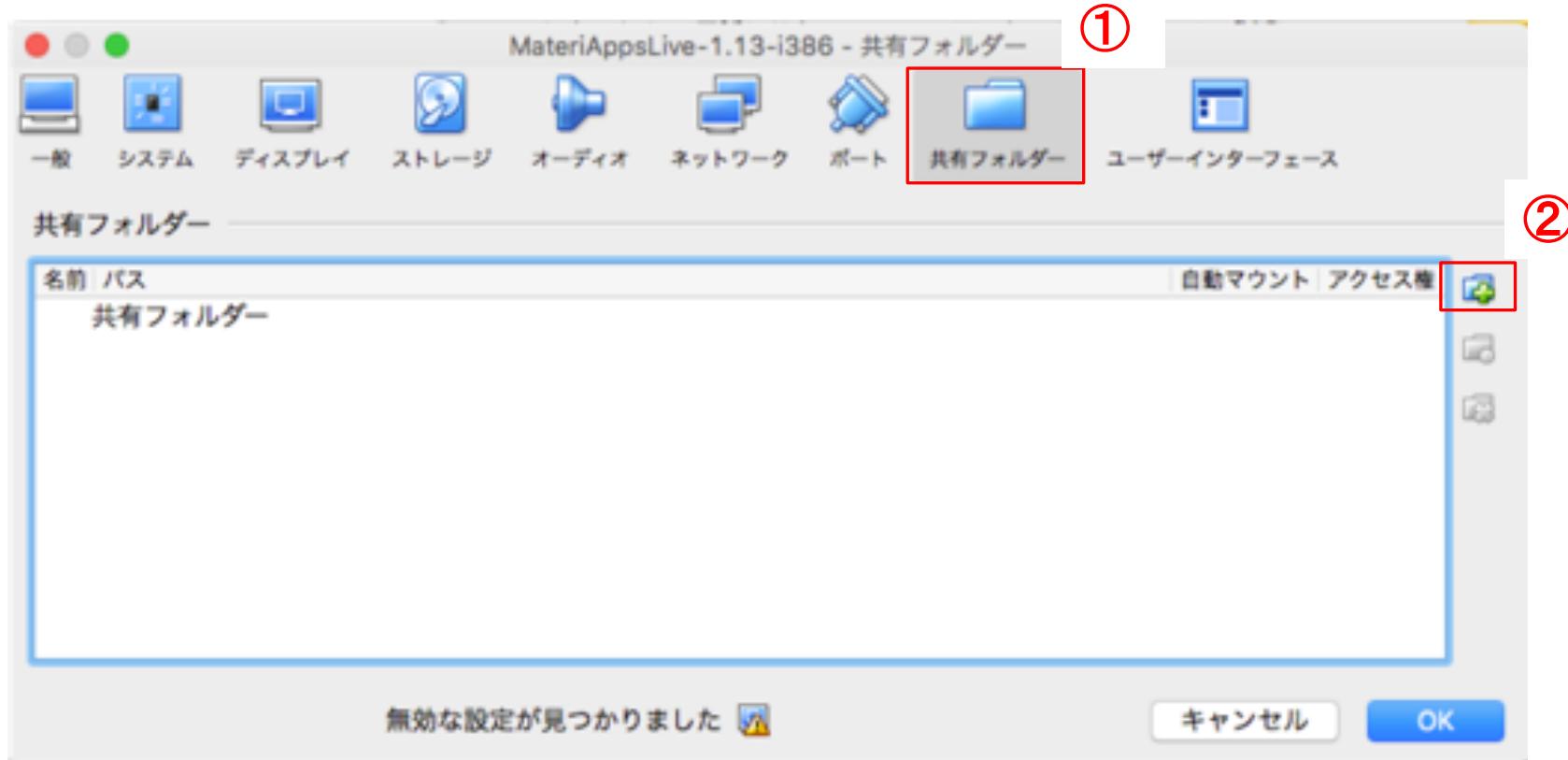
MateriAppsLive-1.13-i386 を選択し、

(注)これは、
MateriAppsLive-1.13-i386 版を用いる場合



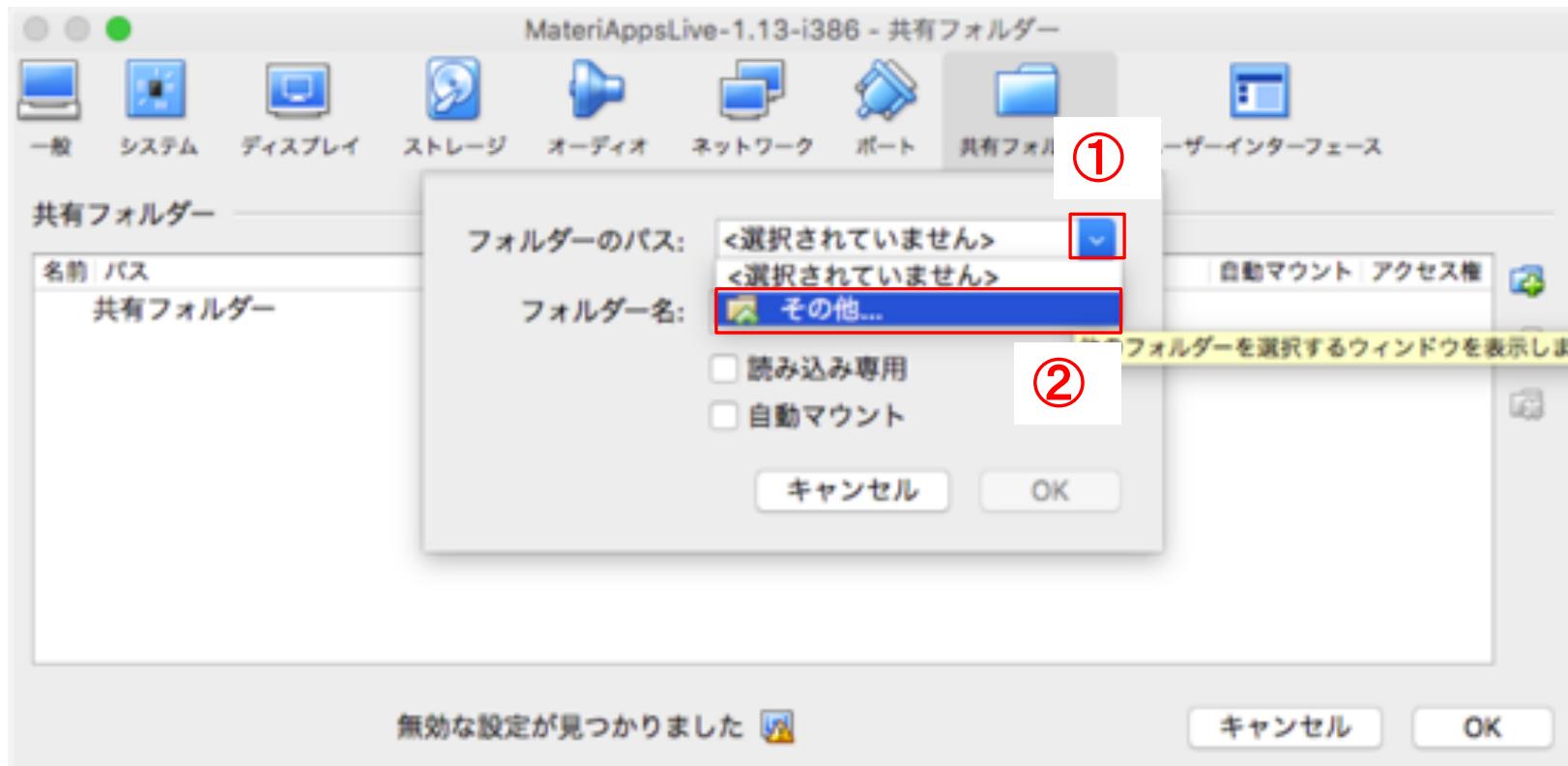
仮想マシンからUSBへのアクセス設定(2)

(2)「共有フォルダー」タブを開き、右側の「+」(新規共有フォルダーを追加します)をクリック



仮想マシンからUSBへのアクセス設定(3)

(3)「フォルダーのパス」の右側の「▼」マークをクリックし、「その他」を選択



仮想マシンからUSBへのアクセス設定(4)

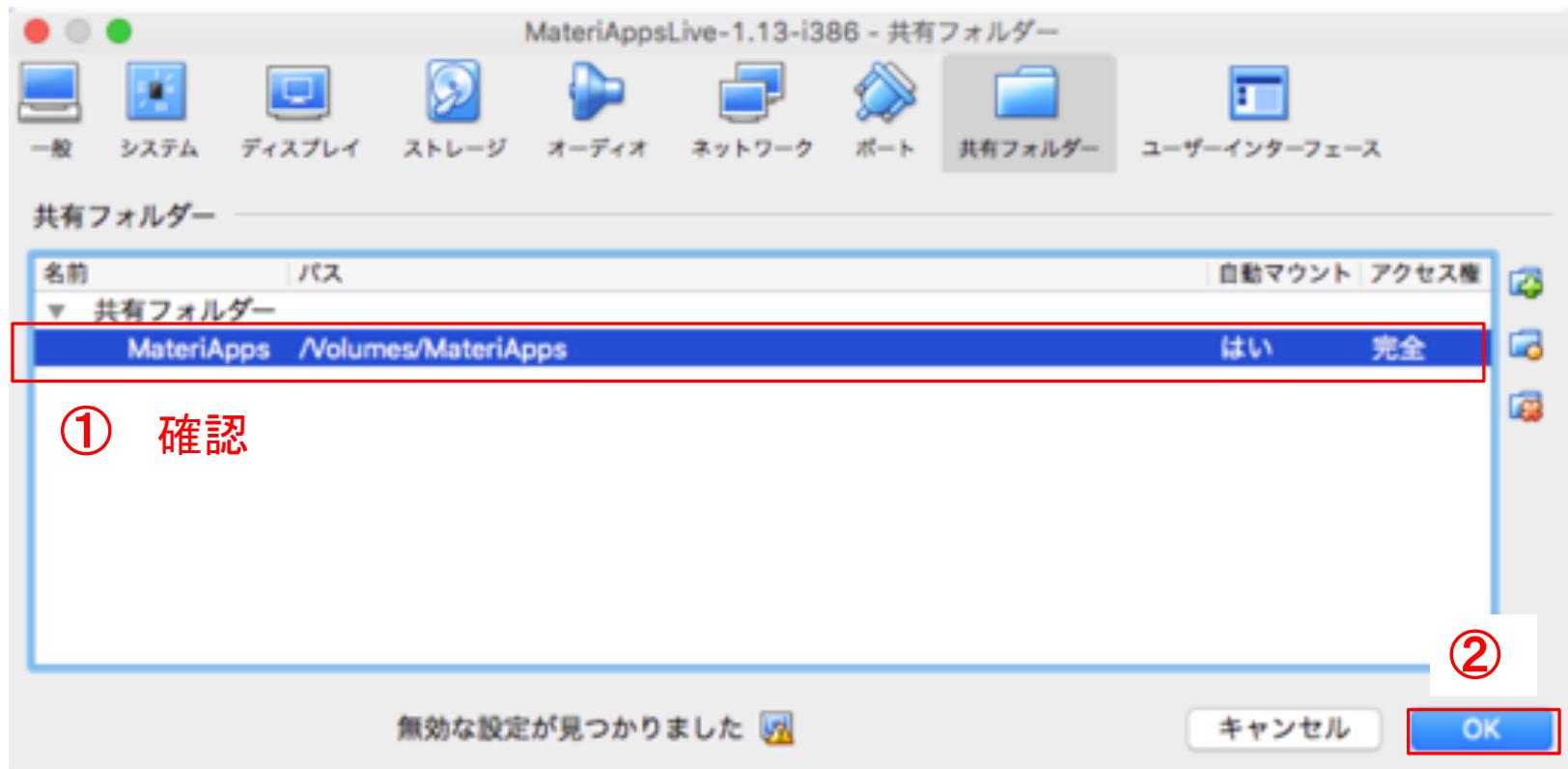
(4) PC内のUSBメモリ(MateriApps)を選択し、「自動マウント」をチェックする。



(注)これは、USBメモリ
のボリュームラベルが
MateriAppsの場合

仮想マシンからUSBへのアクセス設定(5)

(5) 設定を確認し、「OK」ボタンを押す。



(6) 仮想マシンを起動すると、(4)で選択したフォルダが、[/media/sf_MateriApps/](#)の下に見える